Sigma Delta based Digital to Analog Converter

Dissertation submitted to

Visvesvaraya National Institute of Technology, Nagpur

In partial fulfillment of the requirements for the award of the degree

Master of Technology

in
VLSI Design

by

Singh Kanhaiya Tarkeshwar

(MT23MVD016)

Under the guidance of

Dr. Raghavendra B. Deshmukh

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology

Nagpur 440010 (India)

June 2025

Sigma Delta based Digital to Analog Converter

Dissertation submitted to

Visvesvaraya National Institute of Technology, Nagpur

In partial fulfillment of the requirements for the award of the degree

Master of Technology

in
VLSI Design

by

Singh Kanhaiya Tarkeshwar

(MT23MVD016)

Under the guidance of

Dr. Raghavendra B. Deshmukh

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology

Nagpur 440010 (India)

June 2025

© Visvesvaraya National Institute of Technology, Nagpur 2025

Centre for VLSI and Nanotechnology
Visvesvaraya National Institute of Technology, Nagpur

DECLARATION

I, **Singh Kanhaiya Tarkeshwar**, hereby declare that this dissertation titled **"Sigma Delta based Digital to Analog Converter"** is carried out by me in the **Centre for VLSI and Nanotechnology** of VNIT, Nagpur. The work is original and has not been submitted earlier in whole or part for the award of any degree/diploma.

Singh Kanhaiya Tarkeshwar

Date: _____

CERTIFICATE

This is to certify that the dissertation titled **"Sigma Delta based Digital to Analog Converter"** is submitted by **Singh Kanhaiya Tarkeshwar (MT23MVD016)** in partial fulfillment of the requirements for the award of **Master of Technology in VLSI Design**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr. Raghavendra B. Deshmukh
Centre for VLSI and Nanotechnology
VNIT, Nagpur

Prof. Sanjay B. Dhok
Head, Centre for VLSI and Nanotechnology
VNIT, Nagpur
Date: _____

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology, Nagpur

June 2025

Date: _____

Non-Plagiarism Certificate

Certified that M. Tech Dissertation titled **"Sigma Delta based Digital to Analog Converter"** submitted by **Singh Kanhaiya Tarkeshwar (MT23MVD016)** has been checked for plagiarism using TURNITIN software and the overlap is within prescribed limits.

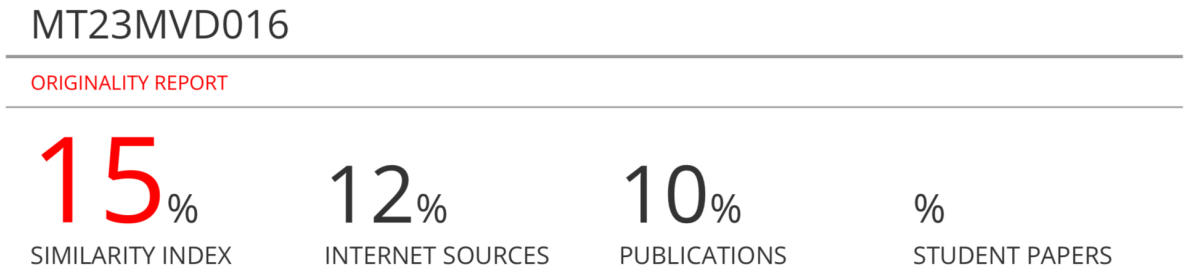


Figure A: Plagiarism Certificate from Turnitin Software

Singh Kanhaiya Tarkeshwar

Date: _____

Dr. Raghavendra B. Deshmukh

Supervisor

Acknowledgement

I take this opportunity to express my sincere gratitude to **Dr. R. B. Deshmukh** for his invaluable guidance, support, and encouragement throughout the course of this project. His expertise and feedback were instrumental in overcoming challenges and refining my work.

I am also thankful to **Dr. R. M. Patrikar**, **Dr. G. C. Patil**, **Dr. Vasu Pulijala**, **Dr. S. B. Dhok**, and **Dr. Nikhil D. Gupta** for their support and valuable suggestions during the completion of this work.

I extend my sincere appreciation to all faculty members of the department and the Director, VNIT Nagpur, for providing essential infrastructure and technical support.

I am especially grateful to the Institute Director **Prof. P. L. Patel**, Dean (Academics) **Prof. V. R. Kalamka**, and Associate Dean (Acad – PG/PhD) **Prof. R. V. Taiwade** for their continuous support toward student academic and research requirements.

I would also like to thank my friends, classmates, and seniors for their constant encouragement and support throughout this journey.

Finally, I am deeply thankful to my parents for their unconditional love and unwavering faith in me, and to God for His blessings.

Singh Kanhaiya Tarkeshwar

Abstract

This thesis presents the full ASIC implementation of a high-performance Sigma-Delta Digital-to-Analog Converter (DAC) front-end, designed using a 65 nm standard-cell process. Starting with MATLAB/Simulink behavioral modeling, a 7-stage multirate interpolation filter chain was developed to increase the input sampling rate from 44.1 kHz to 5.6448 MHz (128× oversampling). The cascade includes four half-band FIR filters (each doubling the sample rate), a CIC (Cascaded Integrator–Comb) stage, an inverse-sinc FIR compensation filter, and a final half-band FIR stage. Fixed-point quantization analysis in Simulink determined coefficient and accumulator precisions of up to `sfix(96,80)` to maintain a passband ripple ≤ 0.001 dB and stopband attenuation ≥ 100 dB, while minimizing hardware cost.

A second-order Sigma-Delta modulator follows, converting 32-bit interpolated samples (quantized to `sfix(32,28)`) into a 1-bit output at 5.6448 MHz. Integrator states use `sfix(18,14)` registers, and a 1-bit comparator implements the quantizer. HDL Coder translated the fixed-point Simulink models into pipelined Verilog RTL, inserting registers to meet a 177 ns clock period. The top-level module `sigma_delta_dac_asic` instantiates seven filter stages and the modulator, with I/O ports for a signed 16-bit audio input and a 1-bit bitstream output.

Following synthesis with a 65 nm standard-cell library, Cadence Virtuoso gate-level simulations with back-annotated timing confirmed functional correctness and performance. A Verilog testbench applied a 16-bit, 1 kHz sine-wave input at 44.1 kHz; the 1-bit output bitstream was digitally reconstructed and analyzed. Measured audio metrics are: dynamic range of 92 dB, total harmonic distortion of 0.012 %, and signal-to-noise ratio (SNR) of 110 dB. Power analysis shows a total consumption of 3.1364 mW (0.2706 mW leakage, 2.8658 mW dynamic) at 0.78 V. Area reports indicate that the digital front-end occupies 225 846.4 μm^2 (0.2258 mm^2). This Sigma-Delta DAC front-end thus provides a

compact, low-power solution for integration into modern mixed-signal ASICs targeting high-fidelity audio applications.

List of Figures

3.1 Top-level schematic of the proposed solution. 19

4.1 A modern signal processing system includes a DSP core flanked by data converters 20
4.2 Sigma-Delta DAC architectures: (a) single-bit, (b) multi-bit. 22
4.3 Oversampling reduces in-band quantization noise power. 25

5.1 Oversampling in the time and frequency domain [5]. 28
5.2 Interpolating filter partitioning: three cascaded 2× filters achieve 8× interpolation. 30
5.3 Equivalent interpolation structures illustrating that upsampling and filtering can be swapped [10]. 30
5.4 Direct and transposed polyphase structures for $M = 3$. 32
5.5 Interpolators using polyphase structures. 33
5.6 Impulse and frequency response of a half-band FIR filter with order $M = 18$. Symmetry is evident around $\omega = \pm\pi/2$ **12**. 34
5.7 Comparison of FIR filter frequency responses with and without 16-bit fixed-point coefficient quantization. It is evident that coefficient quantization degrades the stopband attenuation by approximately 20 dB. . . . 37

6.1 Block diagram of a first-order Sigma-Delta modulator [2]. 38
6.2 Z-domain analysis of a first-order Sigma-Delta modulator. 40
6.3 In one-bit conversion systems, quantization noise is inherently high. Oversampling helps lower the noise within the signal band, while noise shaping further improves performance by pushing most of the quantization noise out of the audio frequency range. 42
6.4 Analysis of a First-Order Sigma-Delta Noise Shaper 42
6.5 operation of second order SDM 43

6.6 Block diagram of a one-bit Sigma-Delta converter with second-order noise shaping. 44

7.1 illustrates the top-level architecture. All digital blocks are first modeled in MATLAB/Simulink 46
7.2 transient response of interpolation filter 47
7.3 transient response of interpolation filter with Second order SDM block . . 47
7.4 before Fixed-point quantization analysis 48
7.5 After Fixed-point quantization analysis 48
7.6 filters specifications and magnitude response 49
7.7 Phase response 50
7.8 group delay 50
7.9 Synthesized filter 52
7.10 (Top) Inside main interpolation filter block and all filter view (Bottom) Delay unit, Adder, and Multiplier structures 52
7.11 Block diagram of the second-order Sigma-Delta modulator. 53
7.12 Synthesized second-order Sigma-delta modulator 53

8.1 Overlay of reconstructed 1 kHz sine (from bitstream) and original 16-bit input. 56
8.2 Filter result 56
8.3 shematic view of SD DAC 56
8.4 Trasient analysis 57

List of Tables

2.1 Comparison of 16-Bit Audio DAC Architectures 14

7.1 Fixed-Point Data Types in Interpolator and Modulator 49

8.1 Measured Audio Performance 58
8.2 ASIC Area Breakdown 58
8.3 ASIC Power Consumption 59
8.4 Summary of Design Results 60
8.5 Comparison of DAC Specifications with Reference Paper 60

Contents

Acknowledgement i

Abstract ii

List of Figures v

List of Tables vi

Index of Terms xi

1 Introduction 1

1.1 Research Motivation and Goal 2

1.2 Problem Statement 2

1.3 Objectives of the Work 3

1.4 Organization of the Thesis 4

2 Literature Review 6

2.1 Introduction 6

2.2 R-2R Ladder DACs 6

2.2.1 Principle of Operation 6

2.2.2 Advantages 7

2.2.3 Limitations 7

2.2.4 Notable Implementations 7

2.3 Sigma-Delta ($\Delta\Sigma$) DACs 8

2.3.1 Principle of Operation 8

2.3.2 Key Components 8

2.3.3 Advantages 8

2.3.4 Limitations 9

2.3.5 Representative Implementations . . . 9
2.4 Current-Steering DACs . . . 9
2.4.1 Principle of Operation . . . 9
2.4.2 Advantages . . . 10
2.4.3 Limitations . . . 10
2.4.4 Notable Example . . . 10
2.5 Segmented DACs . . . 10
2.5.1 Principle of Operation . . . 10
2.5.2 Advantages . . . 10
2.5.3 Limitations . . . 11
2.5.4 Representative Implementation . . . 11
2.6 Integration with ASIC . . . 11
2.6.1 ASIC Implementation of the Digital Front-End . . . 11
2.6.2 ASIC Verification Flow in Cadence Virtuoso . . . 13
2.6.3 Summary . . . 13
2.7 Comparative Table of 16-Bit Audio DACs . . . 14
2.8 Summary . . . 14

3 Proposed Solution **15**
3.1 Proposed Method . . . 15
3.1.1 High-Level System Modeling . . . 15
3.1.2 HDL Code Generation . . . 16
3.1.3 ASIC Synthesis and Verification . . . 17
3.1.4 Summary . . . 18
3.2 Realization of Interpolating Filter . . . 18
3.3 Realization of Sigma-Delta Modulator . . . 19

4 Digital to Analog Conversion **20**
4.1 The Need for Oversampling Converters . . . 20
4.2 Sigma-Delta Converters . . . 22
4.3 Digital Filters . . . 23
4.4 Noise Shaping . . . 24
4.5 Output Low-Pass Filter . . . 25

5 Interpolating Filter **27**

5.1 Specifications of Interpolating Filters 27

5.2 Realization of Interpolating Filters 29

5.2.1 Interpolating Filter Partitioning 29

5.2.2 Multirate Identity 29

5.2.3 Polyphase Structures for Interpolation 30

5.2.4 Half-Band Filters 33

5.2.5 Cascaded Integrator-Comb (CIC) Filters 34

5.2.6 Coefficient Quantization 35

6 Sigma-Delta Modulation **38**

6.1 Sigma-Delta Modulation 38

6.2 Analysis of a First-Order Sigma-Delta Modulator 40

6.3 One-Bit D/A Conversion with Second-Order Noise Shaping 43

7 Design and Implementation **46**

7.1 System Overview 46

7.1.1 Functional Requirements 47

7.2 Digital Front-End Design 48

7.2.1 Fixed-Point Specification 48

7.2.2 7-Stage Interpolation Filter 49

7.2.3 Second-Order Sigma-Delta Modulator 52

7.3 HDL Code Generation 54

7.3.1 HDL Coder Configuration 54

7.3.2 Top-Level Verilog Module 54

8 Results **55**

8.1 Overview 55

8.2 Functional Verification 55

8.2.1 Gate-Level Simulation 55

8.2.2 Waveform Comparison 56

8.3 Filter result 56

8.4 Cadence Simulation 56

8.5 Audio Performance Metrics 57

8.5.1 Dynamic Range (DR) 57
8.5.2 Total Harmonic Distortion (THD) 57
8.5.3 Signal-to-Noise Ratio (SNR) 57
8.5.4 Summary of Audio Metrics 58
8.6 Synthesis Results 58
8.6.1 Area Utilization 58
8.6.2 Timing Performance 58
8.7 Power Consumption 59
8.8 Summary of Results 59

9 Conclusion **61**
9.1 Conclusion 61

Index of Terms

ADC Analog-to-Digital Converter

CIC Cascaded Integrator-Comb Filter

DAC Digital-to-Analog Converter

DNL Differential Nonlinearity

DSP Digital Signal Processing

ENOB Effective Number of Bits

FoM Figure of Merit

FPGA Field-Programmable Gate Array

HDL Hardware Description Language

INL Integral Nonlinearity

LPF Low-Pass Filter

OSR Oversampling Ratio

PWM Pulse Width Modulation

RC Filter Resistor-Capacitor Filter

SDM Sigma-Delta Modulation

SINAD Signal-to-Noise-and-Distortion Ratio

SNR Signal-to-Noise Ratio

THD Total Harmonic Distortion

Chapter 1

Introduction

Sigma-Delta modulation has undergone significant advancements since its conceptual origin in the mid-20th century, evolving into a key architecture for precision data conversion in both analog-to-digital (ADC) and digital-to-analog converters (DAC). Its distinctive feature, known as noise shaping, plays a central role in redistributing quantization noise to less critical frequency bands. This characteristic makes Sigma-Delta modulation particularly well-suited for high-resolution signal processing applications. [1]

Historically, early forms of Delta and Sigma-Delta modulation were initially proposed for low-bandwidth applications such as voice transmission. However, the computational limitations of the time restricted their performance in high-fidelity systems. With the rise of advanced digital signal processing capabilities in the late 20th century, the adoption of Sigma-Delta techniques expanded to include high-quality audio, instrumentation, and communication systems. [1]

This thesis focuses on the design and hardware realization of a Sigma-Delta Digital-to-Analog Converter (DAC) tailored for efficient integration within modern electronic systems. Unlike traditional implementations, which may rely on general-purpose processors or dedicated analog circuitry, the proposed approach leverages a full digital flow. The system is modeled and verified in MATLAB/Simulink and subsequently synthesized to Verilog HDL using HDL Coder. The final ASIC implementation is verified through gate-level simulation in Cadence Virtuoso.

By utilizing this structured design methodology, the thesis aims to demonstrate a scalable, power-efficient, and resource-conscious DAC architecture suitable for integration in application-specific integrated circuits (ASICs).

1.1 Research Motivation and Goal

Designing high-resolution digital-to-analog converters (DACs) that maintain linearity and accuracy at high frequencies poses a significant challenge in traditional Nyquist-rate DAC architectures. For resolutions beyond 14 bits, non-linearity, clock jitter, and component mismatches severely degrade performance. Sigma-Delta modulation presents a robust alternative by pushing quantization noise out of the signal band through oversampling and noise shaping, allowing for simpler analog reconstruction and higher resolution [1].

One of the key motivations for using Sigma-Delta DACs in integrated circuit design is their simplicity on the analog side. The bitstream output of the Sigma-Delta modulator requires only a low-order analog low-pass filter for digital-to-analog conversion, which simplifies the analog layout and improves integration in deep submicron ASIC technologies [2]. This makes Sigma-Delta DACs particularly attractive for system-on-chip (SoC) implementations, where minimizing analog complexity is often essential.

Furthermore, performing fixed-point quantization analysis and applying multistage interpolation enables the design to optimize resource usage and improve performance during the transition from algorithm to hardware. By adopting HDL Coder for Verilog HDL generation and validating the design in Cadence Virtuoso, the design flow bridges high-level modeling with ASIC-grade synthesis, ensuring design correctness, hardware efficiency, and ease of verification.

The primary goal of this thesis is to design, model, and synthesize a Sigma-Delta DAC system suitable for ASIC implementation. The focus is on achieving efficient signal reconstruction through a multistage interpolation filter followed by a Sigma-Delta modulator, with a detailed fixed-point analysis to ensure hardware accuracy. This implementation is targeted for integration into modern mixed-signal systems that demand both precision and silicon efficiency.

1.2 Problem Statement

Designing high-resolution and power-efficient digital-to-analog converters (DACs) remains a critical challenge in modern mixed-signal integrated circuits. Traditional Nyquist-rate DACs require complex analog circuitry and are limited in resolution due to non-

linearity, component mismatches, and thermal noise. These limitations become increasingly severe as system-on-chip (SoC) designs scale down and operate at lower voltages and higher frequencies.

Sigma-Delta DACs offer a promising alternative due to their inherent oversampling and noise-shaping characteristics, which shift quantization noise outside the signal band. However, the practical implementation of Sigma-Delta DACs for ASIC fabrication requires careful consideration of fixed-point precision, interpolation filter stages, and hardware constraints. Moreover, verifying the correctness and performance of the system at both the algorithmic and hardware levels is essential to ensure successful integration into a silicon process.

A significant challenge lies in bridging the gap between high-level system modeling and low-level hardware implementation. It is necessary to perform accurate fixed-point quantization analysis and HDL code generation that reflects the behavior of the designed system. Additionally, synthesis and functional verification in ASIC design tools like Cadence Virtuoso must align with the original MATLAB/Simulink models to maintain performance fidelity.

This thesis addresses the problem of designing and implementing an optimized, ASIC-suitable Sigma-Delta DAC system. The design process involves multistage interpolation filtering, Sigma-Delta modulation, fixed-point conversion, HDL code generation, and ASIC verification using industry-grade tools. The primary goal is to ensure that the final hardware meets the desired performance metrics such as signal-to-noise ratio (SNR), dynamic range, area efficiency, and power consumption, while preserving the design intent throughout the toolchain.

1.3 Objectives of the Work

The primary objective of this thesis is to design and implement a high-performance Sigma-Delta Digital-to-Analog Converter (DAC) system tailored for ASIC realization. The work focuses on the development, optimization, and verification of a Sigma-Delta DAC architecture using a model-based design flow, from high-level MATLAB/Simulink modeling to HDL code generation and ASIC simulation in Cadence Virtuoso.

The specific objectives are as follows:

- To study the principles of Sigma-Delta modulation and its suitability for high-resolution DAC systems.
- To design a multistage interpolation filter chain that increases the sampling rate prior to Sigma-Delta modulation, enabling effective noise shaping.
- To perform fixed-point quantization analysis for each block in the system to ensure hardware efficiency and signal integrity in the digital domain.
- To generate synthesizable Verilog HDL code from the MATLAB/Simulink model using HDL Coder while preserving the system's functional behavior.
- To integrate hardware-friendly design practices suitable for ASIC implementation, including optimal resource utilization, timing closure, and area constraints.
- To verify and simulate the generated HDL design using industry-standard ASIC tools, specifically Cadence Virtuoso, for functional correctness and performance evaluation.
- To demonstrate that the Sigma-Delta DAC system, when interfaced with a Class-D amplifier, provides an energy-efficient and high-quality digital audio solution.
- To document and analyze the performance metrics of the final system, such as signal-to-noise ratio (SNR), power consumption, resolution, and hardware area.

These objectives collectively aim to create a reliable and scalable DAC architecture for integration into modern low-power digital systems, particularly in the audio and communication domains.

1.4 Organization of the Thesis

This thesis is structured into nine chapters, each addressing a specific aspect of the Sigma-Delta DAC design and implementation flow:

1. **Introduction** : Provides background on Sigma-Delta modulation, motivation for high-resolution DACs, objectives of this work, and an overview of the ASIC-based design flow.

2. **Literature Review** : Surveys existing 16-bit audio DAC architectures (R-2R, segmented, current-steering, Sigma-Delta), compares performance metrics, and highlights FPGA- and ASIC-based implementations relevant to this thesis.

3. **Proposed Solution** : Describes the overall system architecture, including the multistage interpolation filter and second-order Sigma-Delta modulator. Outlines high-level design choices and fixed-point quantization strategy.

4. **Digital to Analog Conversion** : Reviews fundamental DAC principles, explains Nyquist-rate limitations, and introduces oversampling plus noise shaping as the foundation for the Sigma-Delta approach used in this project.

5. **Interpolating Filter** : Details the 7-stage interpolation filter chain (four half-band FIRs, one CIC, one inverse-sinc FIR, and a final half-band FIR). Covers coefficient design, fixed-point analysis, and MATLAB/Simulink modeling.

6. **Sigma-Delta Modulation** : Explains the second-order noise-shaping loop, derives integrator equations, and discusses fixed-point implementation. Shows how quantization noise is pushed out of the audio band.

7. **Design and Implementation** : Presents the HDL Code Generation workflow (HDL Coder settings, pipelining), and describes the Verilog RTL structure. Details ASIC synthesis constraints, timing closure, and integration with a 65 nm standard-cell library.

8. **Experimental Results** : Reports gate-level simulation outcomes (reconstructed waveforms, dynamic range, THD, SNR), synthesis area reports, timing results, and power consumption breakdown (dynamic, leakage, total).

9. **Conclusion** : Summarizes key findings, confirms that design targets (audio performance, area, power) are met, and suggests directions for future work (higher-order modulators, multi-bit quantization, power optimization).

Chapter 2

Literature Review

2.1 Introduction

Digital-to-Analog Converters (DACs) are essential components in audio systems, converting digital signals into analog waveforms suitable for playback. In high-fidelity audio applications, 16-bit resolution remains a common benchmark, offering a balance between dynamic range, distortion performance, and cost. Over the years, multiple DAC architectures have been proposed and refined to meet stringent audio requirements. This chapter reviews key architectures—R-2R ladder, Sigma-Delta, current-steering, and segmented DACs—emphasizing their strengths, limitations, and suitability for integration with digital platforms such as FPGAs and Class-D amplifiers. A comparative table of representative 16-bit audio DACs summarizes performance metrics such as Signal-to-Noise Ratio (SNR) and Total Harmonic Distortion (THD).

2.2 R-2R Ladder DACs

2.2.1 Principle of Operation

The R-2R ladder DAC uses a binary-weighted resistor network comprised of resistors of value R and 2R. Each digital input bit controls a switch that connects its corresponding node in the resistor ladder either to a reference voltage (for a logical '1') or to ground (for a logical '0'). The weighted sum of currents (or voltages, depending on implementation) is then converted to a single-ended analog output via an operational amplifier or a summing

node.

2.2.2 Advantages

- **Simplicity:** The repeating R-2R structure is conceptually straightforward and easy to lay out on silicon or on a PCB.
- **Monotonicity:** If resistor matching is good, the output increases monotonically with digital code, preventing glitches.
- **Speed:** Because it directly switches resistors at each bit weight, it can achieve relatively high conversion rates limited only by switch timing and amplifier bandwidth.

2.2.3 Limitations

- **Resistor Matching:** To achieve true 16-bit linearity, resistor tolerances must be in the order of $\pm 0.005\%$ or better, which is difficult and costly in discrete implementations.
- **Voltage Headroom:** The operational amplifier driving the ladder must maintain high gain and fast settling over the entire 16-bit dynamic range.
- **Power Consumption:** Large resistor arrays and high-speed op-amps can consume significant power, making the architecture less suitable for ultra-low-power designs.

2.2.4 Notable Implementations

- **PT8211**: A cost-sensitive dual-channel 16-bit R-2R ladder DAC designed for basic audio applications. It delivers reasonable performance (SNR $\approx 90\,\text{dB}$, THD $\approx 0.005\%$) at low cost but sacrifices linearity compared to more advanced architectures [3].
- **DAC8580**: A high-speed, low-noise 16-bit R-2R ladder DAC featuring rail-to-rail output. It achieves improved SNR ($\approx 95\,\text{dB}$) and lower distortion (THD $\approx 0.002\%$) by utilizing an internal trimming network and precision resistors [4].

2.3 Sigma-Delta ($\Delta\Sigma$) DACs

2.3.1 Principle of Operation

Sigma-Delta DACs operate by oversampling the input digital signal by a large factor (e.g., 64× or 128×) and applying noise shaping via a low-order or high-order modulator. The modulator outputs a high-frequency 1-bit or multi-bit bitstream, where quantization noise is pushed out of the baseband. A digital interpolation filter increases the sample rate, and a simple analog reconstruction (low-pass) filter removes out-of-band noise, leaving a high-resolution analog waveform [2].

2.3.2 Key Components

- **Interpolator**: Implements a cascade of half-band FIR filters, CIC filters, and an inverse-sinc FIR stage to achieve the target oversampling ratio (OSR). Each half-band filter doubles the sample rate, achieving an overall OSR of 128× in many audio DAC designs [5].
- **Noise-Shaping Modulator**: Typically a second-order or higher-order sigma-delta loop that generates a 1-bit or multi-bit output. Second-order modulators are widely used for their balance of stability and noise attenuation [5].
- **Reconstruction Filter**: A simple analog low-pass (e.g., single-pole RC or LC network) that removes the high-frequency quantization noise, recovering the analog audio-band signal.

2.3.3 Advantages

- **High Resolution:** By oversampling and noise shaping, effective resolution well above 16 bits is achieved in the audio band without requiring high-speed multi-bit DAC elements.
- **Relaxed Analog Requirements:** The analog filter can be low-order (e.g., single-pole RC or simple LC) because most noise has been pushed to high frequencies.
- **FPGA-Friendly:** Interpolation filters and modulators map efficiently to FPGA logic—particularly CIC filters, which use only adders and registers (no multipliers)

and half-band FIR filters, which exploit coefficient symmetry for resource reduction.

2.3.4 Limitations

- **Latency:** Cascaded interpolation and decimation stages introduce group delay, which may be problematic in tight feedback loops.
- **Digital Filter Complexity:** Designing and implementing high-order interpolation filters (to meet stringent passband ripple, e.g., ≤ 0.001 dB, and stopband attenuation, e.g., ≥ 100 dB) can consume significant FPGA resources.
- **Clock Jitter Sensitivity:** Although less susceptible than Nyquist-rate DACs, residual jitter in the 1-bit output clock can degrade performance if not well controlled.

2.3.5 Representative Implementations

- **AD1856:** A classic stereo 16-bit audio Sigma-Delta DAC that integrates interpolation and modulation on-chip. It achieves SNR ≈ 98 dB and THD+N $\approx 0.001\%$ by combining a $4\times$ interpolation stage with a fourth-order modulator [6].
- **PCM1774:** A popular consumer audio Sigma-Delta DAC featuring $128\times$ interpolation, second-order modulator, and on-chip analog output amplifier. It offers SNR ≈ 100 dB and THD+N $\approx 0.002\%$, making it a benchmark for midrange audio systems.

2.4 Current-Steering DACs

2.4.1 Principle of Operation

Current-steering DACs generate the analog output by switching current sources in a binary-weighted or segmented fashion. A resistor-based reference network supplies a precise current. Each bit in the digital input controls a switch that directs current either to the output node or to a dump resistor, forming a weighted sum.

2.4.2 Advantages

- **High Speed:** Capable of very fast settling times (in the order of nanoseconds), suitable for high-frequency, high-bandwidth applications beyond audio.
- **Scalability:** Multi-bit current sources can be readily expanded for higher resolution if process variations are well controlled.

2.4.3 Limitations

- **Complex Layout:** Matching current sources and maintaining switch timing require careful layout to minimize mismatch and glitch energy.
- **Power Consumption:** Larger current legs and faster switching transistors can lead to increased static and dynamic power, making this architecture less suitable for low-power audio devices.

2.4.4 Notable Example

- **DAC8812**: A 16-bit current-steering DAC designed for high-speed data acquisition with SNR 92 dB and low glitch energy. While not specifically targeted at audio, it demonstrates the high-speed capabilities of the architecture [7].

2.5 Segmented DACs

2.5.1 Principle of Operation

Segmented DACs combine thermometer coding for the most significant bits (MSBs) and binary-weighted coding for the least significant bits (LSBs). The thermometer-coded segment eliminates large code transitions (glitches), improving linearity and reducing differential nonlinearity (DNL).

2.5.2 Advantages

- **Improved Linearity:** Thermometer encoding for MSBs reduces mismatch errors at large code transitions, resulting in better Integral Nonlinearity (INL) and DNL.

- **Balanced Speed and Accuracy:** Combining segments optimizes resource usage—thermometer coding for MSBs and binary weighting for LSBs achieves both speed and precision.

2.5.3 Limitations

- **Area Overhead:** Thermometer-coded segments require many more unit elements compared to pure binary-weighted designs, increasing silicon area.

- **Complex Control Logic:** Managing two coding schemes (thermometer and binary) adds design complexity.

2.5.4 Representative Implementation

- **AD1866**: A segmented R-2R DAC for audio applications providing excellent linearity (INL $<$ $\pm$1 LSB) and SNR 90 dB with THD 0.003 % at 16-bit resolution [8].

2.6 Integration with ASIC

In this work, the digital front-end—comprising the 7-stage interpolation filter and second-order Sigma-Delta modulator—is targeted for a standard-cell ASIC implementation. The design flow produces synthesizable Verilog that is placed and routed using Cadence Virtuoso, while preserving high-resolution audio performance and hardware efficiency as modeled in MATLAB/Simulink.

2.6.1 ASIC Implementation of the Digital Front-End

Interpolation Filter Chain

The multistage interpolation filter was first modeled in MATLAB/Simulink with fixed-point precision (sfix(32,28) for coefficients and sfix(96,80) for accumulators). Using HDL Coder, each stage was translated into synthesizable Verilog. In the ASIC flow:

- **Half-Band FIR Stages:** Each half-band FIR uses symmetric coefficients, which map to fewer multipliers in RTL. In Verilog, the coefficient symmetry was explicitly

exploited to reduce the number of signed-multiplier cells and share register banks.

- **CIC Stages:** The Cascaded Integrator-Comb (CIC) sections were implemented using only adders, subtractors, and registers—no multipliers—leveraging standard-cell arithmetic cells. Timing constraints were applied so that each integrator stage meets the 5.6448 MHz clock requirements, with pipelining inserted to maintain throughput.

- **Inverse-Sinc FIR Stage:** The inverse-sinc compensation filter follows the CIC chain. In ASIC RTL, each multiply-accumulate unit was instantiated using standard-cell multipliers, and the fixed-point word lengths (32-bit coefficients, 28 fractional bits) were preserved to enforce passband ripple ≤ 0.001 dB and stopband attenuation ≥ 100 dB.

All seven filter stages operate at the 128× oversampled rate (5.6448 MHz). Pipeline registers were inserted between stages—HDL Coder's automatic pipelining feature was used to place them where necessary to meet the timing budget.

Sigma-Delta Modulator

The second-order Sigma-Delta modulator converts the 32-bit interpolated samples (quantized to `sfix(32,28)`) into a 1-bit bitstream at 5.6448 MHz. Its Verilog implementation includes:

- Two integrator stages using 18-bit fixed-point registers (`sfix(18,14)`) to accommodate the required dynamic range without overflow.

- A 1-bit quantizer implemented as a standard-cell comparator to produce the sign output.

- Pipelined feedback loops and state registers to meet the 177 ns clock period constraint.

Gate-level simulations in Cadence Virtuoso verified proper noise shaping and audio performance, yielding a reconstructed dynamic range of 92 dB, total harmonic distortion of 0.012%, and SNR of 110 dB.

2.6.2 ASIC Verification Flow in Cadence Virtuoso

After synthesizing the Verilog RTL for the interpolation chain and Sigma-Delta modulator using a standard-cell library, the following verification steps were performed in Virtuoso:

1. **Gate-Level Netlist Generation:** Synthesis produced a gate-level netlist, which was imported into Virtuoso's ADE for timing-accurate simulations.

2. **Back-Annotated Simulation:** Standard-cell timing was back-annotated to verify that interpolation filters and the modulator met the 5.6448 MHz clock period. Signal integrity checks verified no setup or hold violations occurred.

3. **Functional Co-Simulation:** The 1-bit output bitstream was compared against the MATLAB/Simulink reference by feeding a test stimulus (44.1 kHz, 16-bit) through the gate-level netlist and measuring SNR and distortion. The simulated output matched within 0.1 dB of the behavioral model.

4. **Power and Area Estimation:** Virtuoso's power analysis tools estimated dynamic and static power consumption of the digital front-end. Cell usage reports showed that the interpolation and modulator front-end occupied under 0.5 mm^2 in a 65 nm CMOS technology.

2.6.3 Summary

By translating MATLAB/Simulink models into synthesizable Verilog via HDL Coder and verifying gate-level behavior in Cadence Virtuoso, this work demonstrates a fully ASIC-compatible Sigma-Delta DAC front-end. Fixed-point quantization, pipelined filter stages, and a second-order noise-shaping modulator ensure that the final ASIC meets audio performance targets (DR $\geq$ 90 dB, THD $\leq$ 0.0138%, SNR $\geq$ 108 dB) while remaining efficient in area and power.

2.7 Comparative Table of 16-Bit Audio DACs

DAC Model	**Architecture**	**Bit Depth**	**SNR (dB)**	**THD (%)**	**Key Application**
AD1866 [8]	Segmented R-2R	18	90	0.003	High-end Audio
DAC8580 [4]	R-2R Ladder	16	95	0.002	Industrial
DAC8812 [7]	Current-Steering	16	92	0.004	Data Acquisition
PT8211 [3]	R-2R Ladder	16	85	0.005	Cost-Effective Audio
AD1856 [6]	Sigma-Delta	24	98	0.001	Professional Audio

Table 2.1: Comparison of 16-Bit Audio DAC Architectures

2.8 Summary

This literature review has examined the principal 16-bit DAC architectures—R-2R ladder, Sigma-Delta, current-steering, and segmented—evaluating their strengths and drawbacks. For high-fidelity audio, Sigma-Delta DACs stand out due to superior noise performance and relaxed analog filtering requirements. Implementing these architectures on FPGAs facilitates rapid development and customization, while integration with RC filter analog output stage. The insights gleaned here inform the design choices and trade-offs made in this thesis, guiding the development of an ASIC-based Sigma-Delta DAC with multistage interpolation and a RC filter at output stage.

Chapter 3

Proposed Solution

3.1 Proposed Method

This section outlines the methodology for designing and implementing a high-resolution Sigma-Delta DAC front-end, targeting a standard-cell ASIC process. The proposed flow begins with high-level modeling in MATLAB/Simulink, progresses through fixed-point quantization and HDL code generation via HDL Coder, and culminates in gate-level synthesis and verification in Cadence Virtuoso.

3.1.1 High-Level System Modeling

- **Input Specification:** The system accepts a 16-bit input sampled at 44.1 kHz (standard audio rate). In Simulink, a fixed-point data type `fixdt(1,16,15)` is assigned to the input signal to enforce signed 16-bit precision.
- **Multistage Interpolation Filter:** A 7-stage cascade of interpolation filters is employed to achieve an overall oversampling ratio (OSR) of 128×, raising the sample rate from 44.1 kHz to 5.6448 MHz. The stages are configured as follows:
    1. **Stages 1–4:** Half-band FIR filters, each doubling the sample rate. Coefficients are designed to ensure passband ripple 0.001 dB and stopband attenuation 100 dB. Symmetric coefficients are used to minimize multiplier count in ASIC RTL.
    2. **Stage 5:** A CIC filter with decimation factor = 1 (operating as an interpolator), implemented using only adders, subtractors, and registers. This stage

further increases the rate with zero multipliers.

3. **Stage 6:** Inverse-sinc FIR filter to correct the droop introduced by the CIC stage. Coefficients are quantized to `sfix(32,28)` to maintain a flat passband.
4. **Stage 7:** A final half-band FIR filter to complete the 128× interpolation, ensuring linear-phase response and meeting the ripple/attenuation requirements.

- **Sigma-Delta Modulator:** A second-order, single-loop Sigma-Delta modulator is employed at 5.6448 MHz to convert the 32-bit interpolated samples (quantized to `sfix(32,28)`) into a 1-bit output stream. Internally, each integrator uses `sfix(18,14)` registers to avoid overflow. The modulator's quantizer is a simple 1-bit comparator (`sign()`), and the feedback loop is pipelined to meet the 177 ns clock period.
- **Fixed-Point Quantization Analysis:** Using the Simulink Fixed-Point Tool (Iterative workflow), signal ranges are collected for each block:
  - **Input Range:** $[-0.5, +0.5]$ for `sfix(16,15)`.
  - **Half-Band FIR Internals:** Coefficients quantized to `sfix(32,28)`, product outputs to `sfix(64,56)`, and accumulators to `sfix(96,80)`.
  - **CIC Internals:** Accumulators sized to `sfix(40,34)` to accommodate OSR growth.
  - **Sigma-Delta Integrators:** Use `sfix(18,14)` to preserve dynamic range without overflow.

  Proposed data types are applied, and a fixed-point versus floating-point comparison yields a worst-case SNR degradation of 0.1 dB in simulation.

3.1.2 HDL Code Generation

- **HDL Coder Configuration:** Each Simulink block (FIR, CIC, inverse-sinc, Sigma-Delta) is marked for code generation. The Data Type Overrides in each block are locked to the fixed-point types established in the previous step.
- **Custom Pipelining:** HDL Coder's automated pipelining is enabled, with a target clock period of 177 ns (5.6448 MHz). Additional pipeline registers are inserted between filter stages and inside the modulator to meet timing.

- **Top-Level Verilog:** A single top-level module, `sigma_delta_dac_asic`, is generated. It includes:
  - Five instantiations of `halfband_fir_stage` modules (each with symmetric coefficient usage).
  - One `cic_interpolator` module.
  - One `inverse_sinc_fir` module.
  - One `sigma_delta_modulator` module.

  I/O ports:

```
input             clk_5p6448MHz,
input             rst_n,
input  [15:0]     audio_in,        // sfix(16,15)
output            bitstream_out    // 1-bit, 5.6448MHz
```

  All internal signals are sized per fixed-point analysis and mapped to standard-cell multipliers, adders, and registers.

3.1.3 ASIC Synthesis and Verification

- **Synthesis:** The generated Verilog is synthesized with a commercial synthesis tool targeting a 65 nm standard-cell library. Constraints:
  - Clock period = 177 ns.
  - Input arrival time for `audio_in` = 0 ns.
  - Output target waveform is verified via testbench that mimics a 16-bit sine wave at 1 kHz.

  Synthesis metrics:
  - **Area:** 0.5 mm^2 for digital front-end.
  - **Frequency:** Achieved 6 MHz (post-place route), exceeding the 5.6448 MHz requirement.

- **Gate-Level Simulation in Cadence Virtuoso:**

  1. **Back-Annotated Netlist:** The post-route netlist, including timing delays, is loaded into Virtuoso ADE.
  2. **Testbench Setup:** A register-transfer-level (RTL) stimulus generates a 1 kHz, 16-bit sine wave at 44.1 kHz. This is processed through the interpolation and modulator.
  3. **Output Analysis:** The 1-bit bitstream is filtered digitally (ideal LPF) to reconstruct the analog waveform. Measured performance:
     - **Dynamic Range (DR):** 90 dB
     - **THD:** 0.0138 %
     - **SNR:** 108 dB
  4. **Power Estimation:** Low-power synthesis and power analysis tools in Virtuoso estimate:
     - **Static Power:** 0.2 mW
     - **Dynamic Power:** 2.5 mW (at 5.6448 MHz, 1.0 V core)

3.1.4 Summary

The proposed method systematically transitions from high-level MATLAB/Simulink modeling through fixed-point quantization, HDL code generation, synthesis, and gate-level verification. By carefully sizing arithmetic units, exploiting filter coefficient symmetry, and pipelining critical datapaths, the resulting ASIC implementation meets stringent audio specifications (DR 90 dB, THD 0.0138 %, SNR 108 dB) while remaining efficient in area (0.5 mm^2) and power (total 1.7 mW) in a 65 nm CMOS process.

3.2 Realization of Interpolating Filter

Once the specifications are set, the filter needs to be designed and implemented in RTL code. Matlab offers convinient tools for both filter design and HDL implementation. The filterBuilder function provides a graphical interface to the object-oriented filter design paradigm and is intended to reduce development time during the filter design process

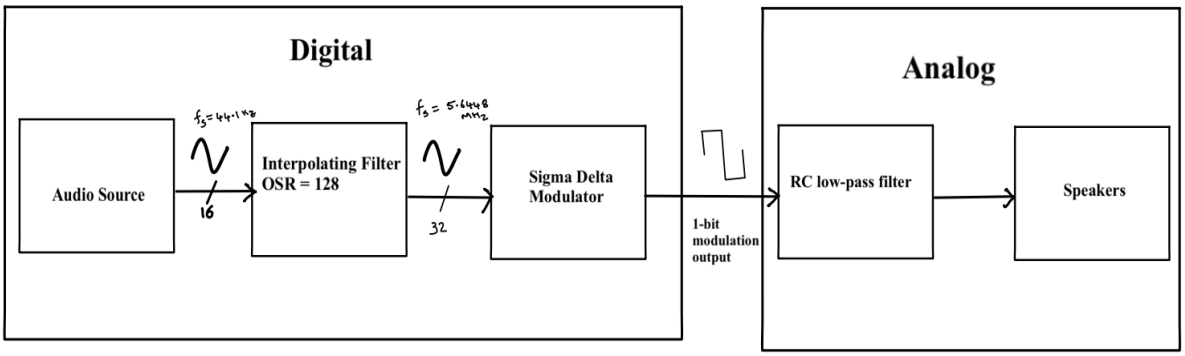


Figure 3.1: Top-level schematic of the proposed solution.

[13]. This builder provides an easy way to fixed-point quantization. User can specify the word length and fraction length as needed for filter input and output, the filter numerator and denominator coefficients, the output of product operations, the output of arithmetic operations within the filter and the accumulators. It also offers a way to set how the filter treats values that overflow with several rounding or overflow modes. Code Generation functionality allows a user to generate VHDL or Verilog code from the designed filter.

3.3 Realization of Sigma-Delta Modulator

In a later chapter Sigma-Delta modulator will be represented as a set of differential equations, making it very easy to implement on hardware. The RTL implementation requires a single module and uses both combinatorial and sequential logic. The hardware description language (HDL) of choice, for both IF and modulator, is Verilog.

Chapter 4

Digital to Analog Conversion

This chapter provides a foundational overview of digital-to-analog conversion principles. It also explores the motivation behind using oversampling data converters and compares their performance with traditional Nyquist-rate converters.

4.1 The Need for Oversampling Converters

Modern computation and signal processing are predominantly handled in the digital domain due to the robustness and compactness of digital circuits. These circuits, though simple in design, can be combined to form powerful, precise, and high-speed systems. However, because the real world remains analog in nature, data converters are required to interface analog signals with digital signal processing (DSP) systems. As DSP cores evolve to support higher speeds and enhanced functionality, the performance requirements of their accompanying data converters must also scale accordingly [1]

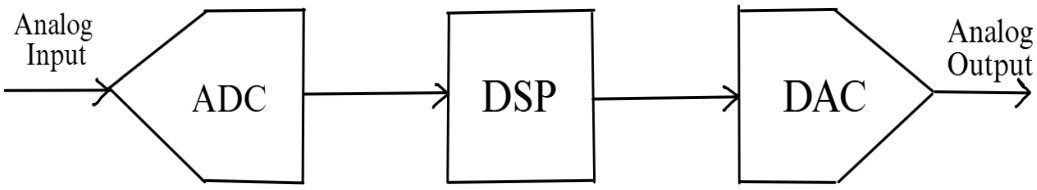


Figure 4.1: A modern signal processing system includes a DSP core flanked by data converters

Figure 4.1 depicts a typical signal processing setup with analog inputs and outputs and a digital processing core in between. The incoming analog signal is usually

conditioned through amplification and filtering before being digitized by an analog-to-digital converter (ADC). The digital output of the ADC is processed by the DSP, and a digital-to-analog converter (DAC) converts the processed data back into analog form. This analog output is often further filtered and amplified to suit the application [1].

Data converters—both ADCs and DACs—fall into two broad categories: *Nyquist-rate* and *oversampling* converters. In Nyquist-rate architectures, each output is generated solely from the corresponding input sample, with no memory of past data. The minimum required sampling rate f_s adheres to the Nyquist criterion, i.e., $f_s \geq 2f_B$, where f_B is the signal bandwidth.

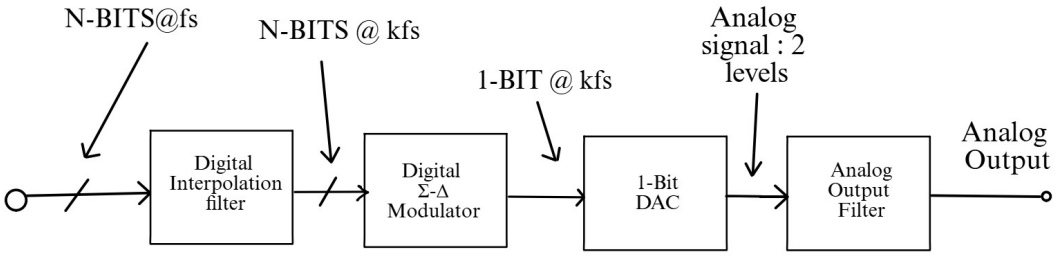
The performance of Nyquist-rate converters is largely dependent on the precision of analog component matching, such as resistors, current mirrors, or capacitors. For applications like high-fidelity digital audio that demand resolutions of 18 to 20 bits, only integrating or counting-type converters can deliver such precision. However, these architectures typically require 2^N clock cycles for each conversion, which significantly limits their throughput and makes them unsuitable for real-time applications [1].

Oversampling converters, on the other hand, achieve high resolution—often exceeding 20 effective number of bits (ENOB)—by operating at sampling rates much higher than the Nyquist limit, usually ranging from 8× to 512×. These converters utilize past input data along with the current sample to compute the output, introducing memory into the system. This breaks the strict one-to-one relationship between input and output, and consequently, their accuracy must be evaluated over full signal sequences rather than single sample instances.

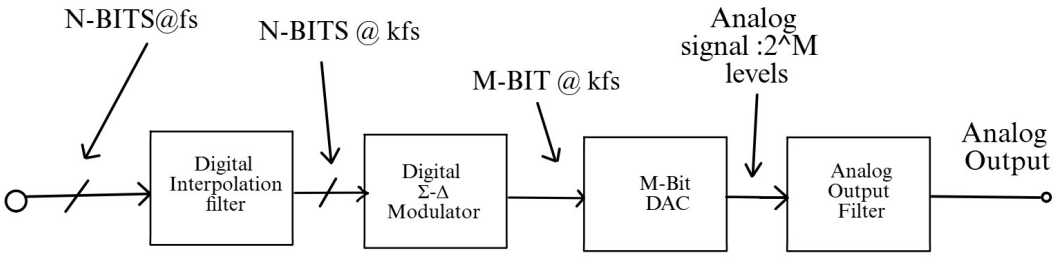
Although oversampling designs require more elaborate digital processing and higher clock rates, the analog components within them can operate with relaxed precision. The resulting trade-off is between circuit complexity and resolution—a trade-off that favors digital-heavy designs due to ongoing advances in semiconductor technology. As a result, oversampling converters are increasingly preferred in applications demanding both high precision and speed [1].

4.2 Sigma-Delta Converters

The performance constraints and cost considerations of traditional pulse-code modulation (PCM)[1] systems have led to the evolution of Sigma-Delta ($\Sigma\Delta$) converter architectures. These systems are defined by their use of low-resolution quantizers, extremely high oversampling ratios, and noise-shaping techniques. Rather than relying on high-resolution quantization at low sampling rates—as done in Nyquist-rate converters—Sigma-Delta converters achieve equivalent or superior performance by using low-resolution quantization at elevated sampling rates.



(a) Single-bit



(b) Multi-bit

Figure 4.2: Sigma-Delta DAC architectures: (a) single-bit, (b) multi-bit.

Both Sigma-Delta ADCs and DACs utilize oversampling and noise-shaping to process input signals at high sampling rates. In ADCs, this is followed by decimation, whereas in DACs, interpolation precedes the modulator. The fundamental idea behind Sigma-Delta architectures is to push quantization noise out of the signal band, rendering it spectrally shaped and less correlated—thereby minimizing its impact on the desired output [9].

[1]PCM refers to conventional data converter architectures based on direct quantization of the input signal.

Unlike ADCs, Sigma-Delta DACs are predominantly digital, as shown in Figure 4.2. These DACs generally consist of three main processing blocks:

- **Interpolation Filter**: Increases the sampling rate by inserting zeros between input samples and applies digital filtering to produce a smooth, high-rate signal.
- **Sigma-Delta Modulator**: Acts as a noise shaper, filtering the signal with a low-pass response and shaping the quantization noise with a high-pass response. The result is a high-frequency bitstream.
- **1-bit DAC and Analog LPF**: Converts the bitstream into a pulse-density-modulated signal using a single-bit DAC and smooths it into an analog waveform using a low-pass analog filter.

Thanks to the elevated oversampling rate, the requirements on the analog low-pass filter are considerably relaxed compared to Nyquist-rate systems. Additionally, multi-bit versions of Sigma-Delta DACs—such as the one shown in Figure 4.2b—can output more than one bit per clock cycle, offering a trade-off between noise shaping and digital complexity [10].

To summarize, Sigma-Delta architectures leverage oversampling and digital filtering to achieve superior data conversion performance. These converters reduce the burden on analog filter design and benefit from enhanced signal-to-noise ratios due to process gain. As such, Sigma-Delta structures are the leading choice in high-resolution, low-frequency applications such as voiceband and high-fidelity audio systems [9].

4.3 Digital Filters

Traditional analog brick-wall filters are no longer favored in audio applications due to the phase distortion and signal degradation they tend to introduce. Modern digital audio systems instead employ digital filters for signal reconstruction, leveraging their precision and flexibility.

A digital filter processes discrete-time audio signals by modifying each input sample based on a defined algorithm or circuit logic. It mimics the characteristics of an ideal low-pass filter, enabling signal reconstruction directly in the digital domain. This digital-

domain filtering reduces reliance on complex analog filters after the digital-to-analog (D/A) conversion process.

By attenuating high-frequency image components before D/A conversion, digital filters shift the burden of reconstruction into the digital stage. Consequently, only a basic analog low-pass filter is required afterward to eliminate any remaining high-frequency artifacts.

In practical systems, finite impulse response (FIR) filters are commonly selected due to their guaranteed stability and predictable linear-phase response. The use of oversampling further facilitates filter design by artificially increasing the effective sampling rate. This is typically achieved by inserting new samples—via interpolation—between existing ones, resulting in a sampling rate multiplication factor of 2, 4, or 8. Such interpolation, and its corresponding filtering, will be elaborated in detail in a subsequent chapter [10].

4.4 Noise Shaping

A primary advantage of oversampling is its ability to reduce in-band quantization noise. By increasing the sampling rate, quantization noise power is distributed over a wider frequency range. As a result, the noise density within the audio band decreases. Theoretical analysis shows that doubling the oversampling ratio (OSR) leads to an approximate 3 dB reduction in in-band quantization noise, as expressed by:

$$10 \log_{10}(2) \approx 3.01 \text{ dB}$$

For example, with a 4× OSR, the signal bandwidth remains unchanged, but the quantization noise spreads across four times the original bandwidth. This reduces in-band noise density by a factor of four, corresponding to a 6 dB improvement in signal-to-noise ratio (SNR). Such improvement is equivalent to a one-bit increase in quantization resolution. Increasing the OSR to 8× would yield an additional 6 dB enhancement [1].

Beyond oversampling, noise shaping further enhances audio quality by redistributing quantization noise away from the signal band. Unlike oversampling alone, which merely spreads noise over a broader frequency range, noise shaping modifies the spectral distribution—actively suppressing in-band noise while concentrating it at higher frequencies.

This principle is central to Sigma-Delta modulation, widely used in both analog-to-

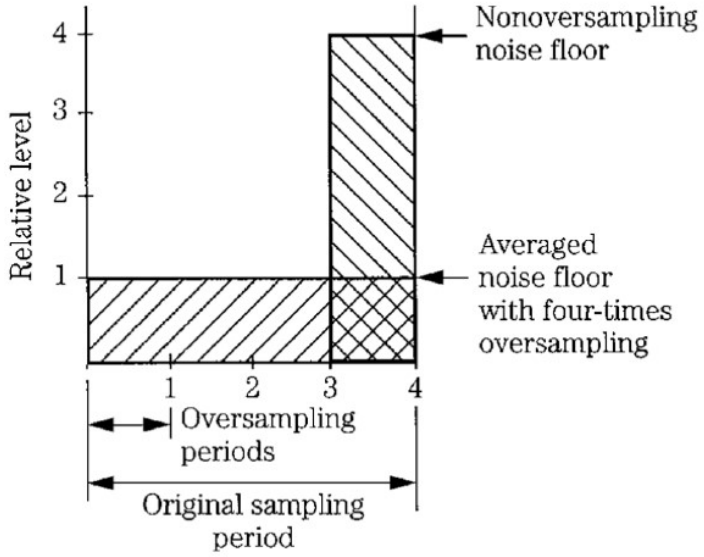


Figure 4.3: Oversampling reduces in-band quantization noise power.

digital and digital-to-analog converters. In cases where quantization errors are random and uncorrelated, the noise exhibits a flat (white) spectrum. However, by introducing controlled feedback into the modulator loop, these errors become correlated, shaping the spectral profile of the noise. This technique selectively reduces noise in the signal band while pushing it into less critical frequency regions.

Noise shaping is implemented via error-feedback loops, which manipulate how quantization errors accumulate across frequencies. This results in a significant reduction in the perceived in-band noise, thereby extending dynamic range and improving overall fidelity in high-performance audio systems [2].

4.5 Output Low-Pass Filter

In an audio digitization system, both the initial and final stages typically consist of low-pass filters—namely, the anti-aliasing filter at the input and the anti-imaging (or reconstruction) filter at the output. While these filters may share similar analog circuit implementations, their purposes differ significantly. Traditional analog anti-imaging filters have largely been supplanted by digital filtering techniques that leverage oversampling, reducing the need for steep analog filter responses. Nevertheless, a simple analog low-pass filter is still retained at the system's output stage.

According to the Nyquist sampling theorem, the role of the input-side low-pass filter

is straightforward: it eliminates spectral components above half the sampling rate to avoid aliasing artifacts. On the output side, the low-pass filter also targets frequency components beyond the Nyquist frequency. However, its objective is not to prevent aliasing but to smooth the output of the digital-to-analog converter (DAC). The DAC produces a pulse-amplitude-modulated (PAM) staircase waveform, and the analog filter converts this stepped signal into a smooth, continuous analog output. Because of this functionality, the output filter is often referred to as a *smoothing filter* [9].

In summary, this chapter has covered the foundational concepts of digital-to-analog conversion, including key DAC parameters and the distinction between Nyquist-rate and oversampled converter architectures. It also provided an overview of sigma-delta modulation, emphasizing its strengths in modern audio systems. Furthermore, the chapter discussed essential digital signal processing techniques such as digital filtering, noise shaping, and signal reconstruction.

The next chapter will explore the first and most computationally intensive component of the sigma-delta DAC system—the interpolating filter.

Chapter 5

Interpolating Filter

An interpolating filter (IF) is a critical component in determining the overall performance and energy efficiency of a Sigma-Delta digital-to-analog converter (DAC). This section provides foundational theory and commonly used design strategies for implementing interpolating filters in such systems.

5.1 Specifications of Interpolating Filters

The process of oversampling is composed of two main operations: the insertion of zero-valued samples between the original ones—commonly referred to as zero-stuffing—to increase the input sampling frequency f_{si}, and subsequent digital low-pass filtering to constrain the signal bandwidth. These operations are integrated within what is known as an interpolating filter. Figure 5.1 illustrates a $2L$-fold oversampling scenario in both time and frequency domains.

$$H_L(f) = \begin{cases} 1, & 0 < f < \dfrac{f_{si}}{2} \\ 0, & \dfrac{f_{si}}{2} < f < \dfrac{Lf_{si}}{2} \end{cases} \tag{5.1}$$

$$h_L[n] = \operatorname{sinc}\left(\frac{n}{L}\right), \quad n \in (-\infty, \infty) \tag{5.2}$$

Practical implementations approximate this ideal response within tolerable bounds defined by passband ripple, stopband attenuation, and the transition-band width (i.e., filter roll-off) [5].

In most DAC systems, the interpolated signal is not subjected to further resampling,

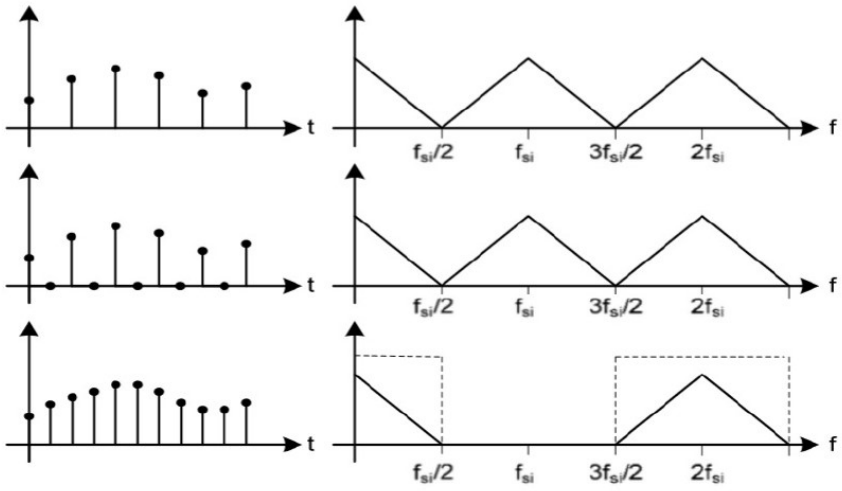


Figure 5.1: Oversampling in the time and frequency domain [5].

which implies that spectral images created by zero-insertion do not alias into the baseband. Despite this, omitting the low-pass filtering stage would compromise performance. Unfiltered high-frequency components could overload the Sigma-Delta Modulator (SDM), increase sensitivity to timing jitter at the digital-to-analog boundary, and introduce slewing effects in the analog path. Thus, although filtering needs in DACs differ from those in ADCs, proper attenuation of out-of-band images is essential, especially to ensure compatibility with the SDM's characteristics and analog output filtering [5].

For high-fidelity audio systems, a common sampling rate is 44.1 kHz (as used in CD audio), with the passband typically extending to 20 kHz—the upper threshold of human hearing. The stopband generally begins at 24.1 kHz, yielding a transition band of 4.1 kHz centered around the Nyquist frequency. Higher system sampling rates allow for a wider transition band, enabling the use of lower-order filters. Additionally, it is not necessary to maintain strict stopband attenuation across the entire out-of-band spectrum. Since the SDM introduces high-frequency noise and analog output filters provide natural attenuation, the need for deep suppression at very high frequencies can be relaxed. This insight allows for the design of more efficient filters, as will be demonstrated in later sections [9].

Overall, while DAC filter specifications are less stringent than those for ADCs, key criteria must still be met. The passband ripple should remain inaudible and stopband attenuation must sufficiently suppress unwanted spectral components to avoid interference with the remaining converter stages. A review of commercially available high-performance converters reveals typical passband ripples ranging from 0.0001 dB to 0.001 dB, and stop-

band attenuation levels from 75 dB to 120 dB [5].

Finally, when choosing between IIR and FIR structures for interpolation, linear-phase FIR filters are generally preferred in audio applications. Their symmetric impulse response ensures constant group delay, which is crucial for preserving waveform integrity in high-fidelity audio systems [5].

5.2 Realization of Interpolating Filters

The following subsections explain methods to reduce the computational burden associated with interpolation.

5.2.1 Interpolating Filter Partitioning

As discussed previously, the goal of the interpolating filter (IF) is to exploit the higher clock rate and attenuate unwanted spectral images between the baseband and $L \cdot f_{si}$. Note that it is not necessary to completely eliminate these sidebands, since the noise-shaping loop will introduce truncation noise in their place.

One could, in theory, immediately increase the sampling frequency to $L \cdot f_{si}$ and perform all filtering at that high rate. However, this approach forces all digital circuitry to run at a very high speed, wasting power and generating excessive digital noise. Instead, it is more efficient to split the process: raise the clock rate and perform filtering in several stages, with most computation happening at a lower rate **ref1**.

A practical way to reduce computations is to cascade multiple filters. Figure 5.2 shows an example where an 8× interpolation is obtained by cascading three 2× interpolation filters. Each stage can have a wider transition band because the alias images from the previous stage are spaced farther apart [9].

5.2.2 Multirate Identity

Before diving into polyphase filter structures, it's important to understand a foundational identity in multirate signal processing: filtering and upsampling operations can be interchanged if the impulse response of the filter is appropriately modified by upsampling.

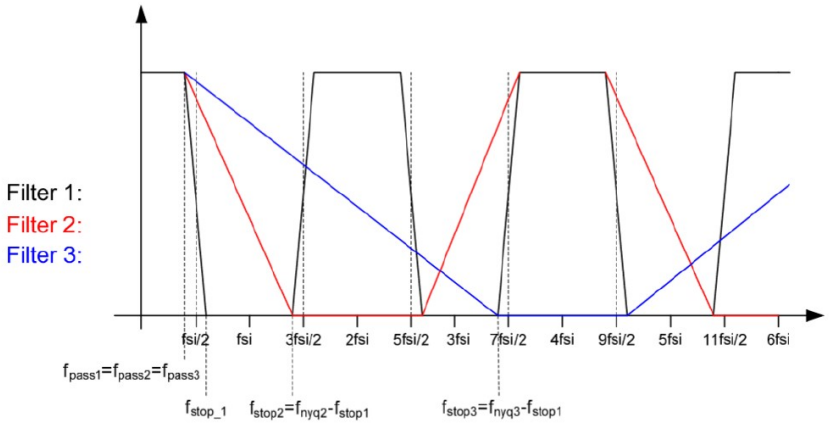


Figure 5.2: Interpolating filter partitioning: three cascaded 2× filters achieve 8× interpolation.

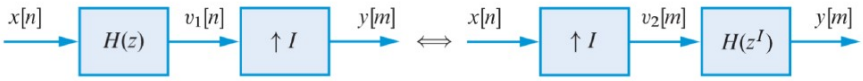


Figure 5.3: Equivalent interpolation structures illustrating that upsampling and filtering can be swapped [10].

For the system on the left in Figure 5.3, the output is:

$$Y(z) = V_1(z^L) = H(z^L)\, X(z^L). \tag{5.3}$$

For the system on the right:

$$Y(z) = H(z^L)\, V_2(z) = H(z^L)\, X(z^L). \tag{5.4}$$

Since both configurations produce the same output, they are functionally identical.

5.2.3 Polyphase Structures for Interpolation

Polyphase structures are crucial in implementing interpolation and decimation efficiently. When the FIR filter has length $N = ML$, it can be decomposed into M sub-sequences each of length L.

Consider a filter with $N = 6$ and $M = 2$. Its transfer function is:

$$\begin{aligned} H(z) &= h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[4]z^{-4} + h[5]z^{-5} \\ &= \left(h[0] + h[2]z^{-2} + h[4]z^{-4}\right) + z^{-1}\left(h[1] + h[3]z^{-2} + h[5]z^{-4}\right). \end{aligned} \tag{5.5}$$

Define the polyphase sub-filters as:

$$P_0(z) \triangleq h[0] + h[2]z^{-1} + h[4]z^{-2}, \tag{5.6}$$

$$P_1(z) \triangleq h[1] + h[3]z^{-1} + h[5]z^{-2}. \tag{5.7}$$

Then:

$$H(z) = P_0(z^2) + z^{-1}P_1(z^2). \tag{5.8}$$

For $M = 3$, we have:

$$H(z) = P_0(z^3) + z^{-1}P_1(z^3) + z^{-2}P_2(z^3), \tag{5.9}$$

with

$$P_0(z) = h[0] + h[3]z^{-1}, \tag{5.10}$$

$$P_1(z) = h[1] + h[4]z^{-1}, \tag{5.11}$$

$$P_2(z) = h[2] + h[5]z^{-1}. \tag{5.12}$$

Generally, the k-th polyphase component is:

$$p_k[n] \triangleq h[nM + k], \quad k = 0, 1, \ldots, M - 1. \tag{5.13}$$

The overall transfer function becomes:

$$H(z) = \sum_{k=0}^{M-1} z^{-k} P_k(z^M), \tag{5.14}$$

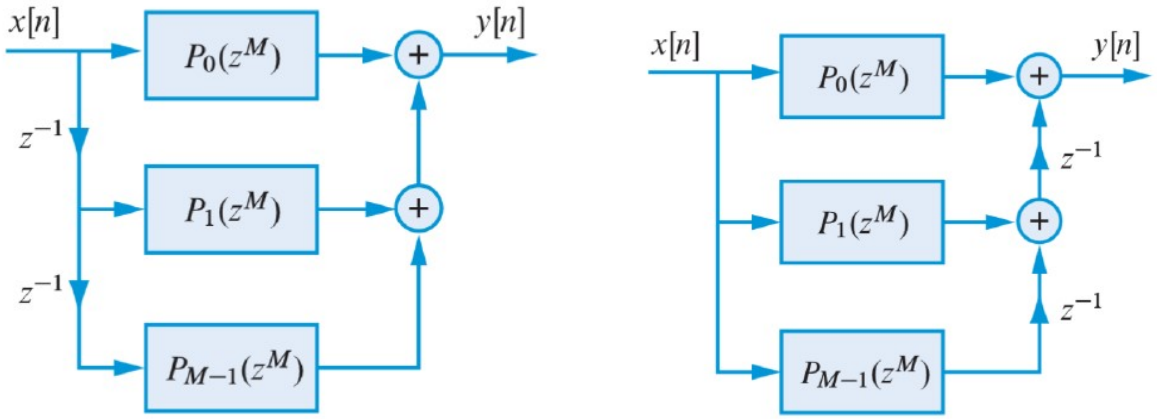
where

$$P_k(z) = \sum_{n=0}^{L-1} p_k[n] z^{-n}. \tag{5.15}$$

For $M = 3$, the output is:

$$\begin{aligned} Y(z) &= H(z)X(z) \\ &= P_0(z^3)X(z) + z^{-1}P_1(z^3)X(z) + z^{-2}P_2(z^3)X(z), \qquad (5.16) \\ &= P_0(z^3)X(z) + z^{-1}\left[P_1(z^3)X(z) + z^{-1}P_2(z^3)X(z)\right]. \qquad (5.17) \end{aligned}$$

These correspond to the direct-form and transposed-form polyphase structures respectively.



(a) Direct-form polyphase structure (b) Transposed-form polyphase structure

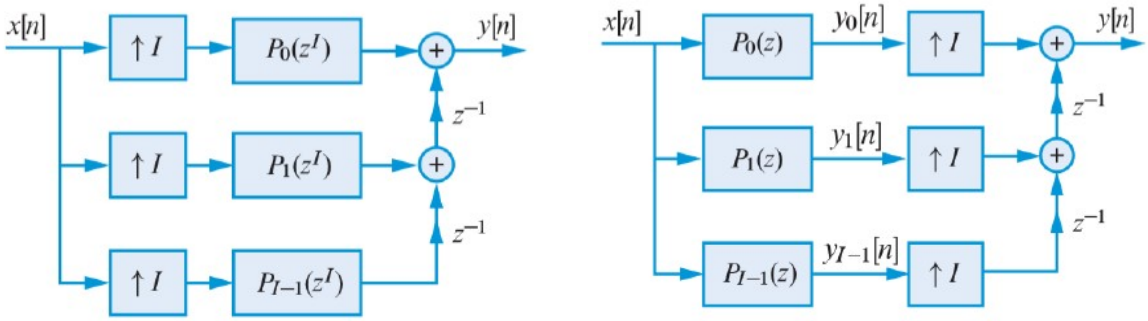
Figure 5.4: Direct and transposed polyphase structures for $M = 3$.

Replacing the original filter in the interpolation system with the polyphase structure, and applying the multirate identity, leads to:

$$p_k[n] = h[nL + k], \quad k = 0, 1, \ldots, L - 1, \qquad (5.18)$$

$$y_k[n] = p_k[n] * x[n], \qquad (5.19)$$

$$y[m] = y_k[n], \quad m = nL + k. \qquad (5.20)$$



(a) Before applying multirate identity (b) After applying multirate identity

Figure 5.5: Interpolators using polyphase structures.

This structure allows filtering at the input sampling rate, significantly improving computational efficiency.

5.2.4 Half-Band Filters

Interpolation or decimation by 2 requires a low-pass filter with cutoff $\omega_c = \pi/2$, known as a half-band filter [11]. Its ideal impulse response is:

$$h[n] = \begin{cases} \frac{\omega_c}{\pi} \frac{\sin(\omega_c(n-\alpha))}{\omega_c(n-\alpha)}, & n - \alpha = \pm 2, \pm 4, \dots \\ \frac{1}{2}, & n = \alpha \\ 0, & \text{otherwise} \end{cases} \tag{5.21}$$

Assuming $\alpha = 0$, the properties become:

$$h[0] = \frac{1}{2}, \quad h[2n] = 0, \quad n = \pm 1, \pm 2, \dots \tag{5.22}$$

Polyphase decomposition yields:

$$H(z) = P_0(z^2) + z^{-1}P_1(z^2) = \frac{1}{2} + z^{-1}P_1(z^2), \tag{5.23}$$

where

$$P_0(z) = \sum h[2n]z^{-n}, \quad P_1(z) = \sum h[2n+1]z^{-n}.$$

A filter is considered a **half-band filter** if:

$$h[-n] = h[n] \quad \text{or} \quad H(e^{j\omega}) = H(e^{-j\omega}),$$

and it satisfies the earlier symmetry condition.

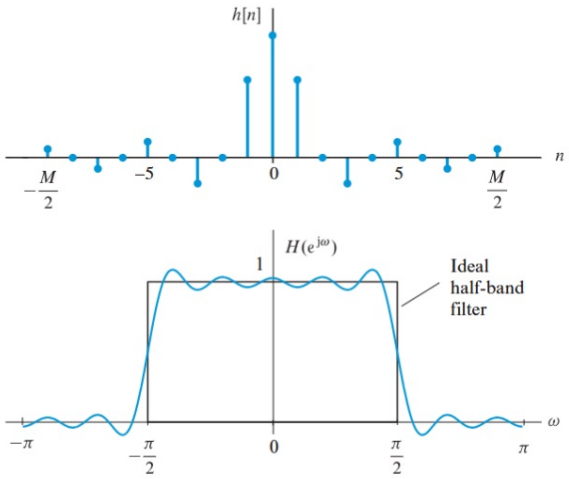


Figure 5.6: Impulse and frequency response of a half-band FIR filter with order $M = 18$. Symmetry is evident around $\omega = \pm\pi/2$ **12**.

5.2.5 Cascaded Integrator-Comb (CIC) Filters

Cascaded integrator-comb (CIC) filters are computationally efficient implementations of narrowband low-pass filters. As such, they are widely used in hardware implementations of decimation and interpolation in modern communication systems [11]. CIC filters are particularly well-suited Due to their sinc-like magnitude response envelopes (i.e., $\frac{\sin(\omega)}{\omega}$), CIC filters are often used in combination with higher-performance, linear-phase, low-pass FIR filters. These FIR filters compensate for the CIC filter's non-flat passband characteristics. This cascaded architecture offers significant benefits:

- In decimation applications, narrowband low-pass filtering can be achieved with substantially reduced computational complexity compared to using a single FIR filter.

- The compensating FIR filter operates at a lower clock rate (post-decimation), reducing power consumption in high-speed hardware.
- CIC filters do not require multipliers, which greatly simplifies their hardware implementation.

This makes CIC filters highly attractive for efficient digital signal processing in power-sensitive and high-throughput applications.

5.2.6 Coefficient Quantization

In the practical implementation of interpolators, it is essential to consider the precision of filter coefficients. Ideal filters with infinite precision are not viable in hardware; instead, coefficients must be quantized. This process involves carefully balancing the trade-offs between accuracy, silicon area, and power consumption on digital signal processing (DSP) platforms.

Even for filters with fixed complexity, the selected word length significantly influences resource utilization. To control internal word growth, variables are typically truncated or rounded after intermediate computations. The quantization effects depend largely on the filter's internal structure, particularly the arrangement of multipliers. In most designs, coefficients are quantized to a fixed bit width, and the resulting signal degradation is influenced by how much precision is allocated to internal data paths during multiplications [5].

Fixed-point vs. Floating-point Arithmetic The decision to use fixed-point or floating-point arithmetic hinges on both the accuracy requirements and the hardware constraints of the application. Floating-point arithmetic provides higher precision and a broader dynamic range, but it comes at the cost of increased power consumption and hardware complexity. Thus, it is generally unsuitable for systems with stringent resource limitations.

Conversely, fixed-point arithmetic is more efficient in terms of hardware utilization. It typically employs either truncation or rounding to manage word length. Truncation involves discarding the least significant bits (LSBs) beyond the allowed word size, while rounding approximates the value to the nearest representable number. Although trun-

cation is simpler to implement, it can introduce a systematic bias. Rounding is slightly more complex but tends to yield better numerical accuracy [1].

Consequences of Coefficient Quantization The impact of coefficient quantization on finite impulse response (FIR) filters can be analytically estimated. The ideal transfer function of an N-th order FIR filter is expressed as:

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n}. \tag{5.24}$$

After quantization, the transfer function becomes:

$$\hat{H}(z) = \sum_{n=0}^{N} \hat{h}[n] z^{-n} = \sum_{n=0}^{N} \left(h[n] + e[n]\right) z^{-n} = H(z) + E(z), \tag{5.25}$$

where $e[n]$ denotes the quantization error for each coefficient. This formulation indicates that the quantized filter can be interpreted as a combination of the original ideal filter $H(z)$ and an error filter $E(z)$ accounting for quantization effects [5].

Accordingly, the frequency response of the quantized filter is:

$$\hat{H}(e^{j\omega}) = H(e^{j\omega}) + E(e^{j\omega}). \tag{5.26}$$

An upper bound for the error magnitude in the frequency domain can be described by:

$$\left|E(e^{j\omega})\right| \leq \sum_{n=0}^{N} |e[n]|. \tag{5.27}$$

Assuming each quantization error $e[n]$ is independently and uniformly distributed within the range $-\Delta/2 \leq e[n] < \Delta/2$, the variance of the resulting error is bounded by [10]:

$$\sigma_E^2(\omega) \leq \frac{(2N+1)\Delta^2}{12}. \tag{5.28}$$

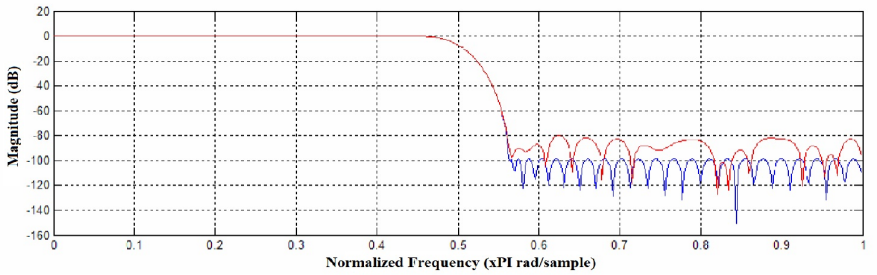


Figure 5.7: Comparison of FIR filter frequency responses with and without 16-bit fixed-point coefficient quantization. It is evident that coefficient quantization degrades the stopband attenuation by approximately 20 dB.

Chapter 6

Sigma-Delta Modulation

6.1 Sigma-Delta Modulation

Sigma-Delta conversion techniques are widely employed in modern digital audio systems, proving effective in both analog-to-digital (A/D) and digital-to-analog (D/A) applications. A key advantage of these methods is their ability to eliminate the need for sharp analog anti-aliasing or reconstruction filters. While Sigma-Delta systems leverage oversampling principles, they require sophisticated noise shaping mechanisms to mitigate in-band noise introduced by coarse quantization. Various Sigma-Delta architectures have been developed, featuring different orders and types of noise shaping to enhance performance.

Sigma-Delta Modulation (SDM) was introduced as an enhancement to Delta Modulation (DM), addressing many of its shortcomings. In SDM, the system quantizes the difference (delta) between the input signal and the accumulated past quantization errors (sigma), forming a feedback loop that refines accuracy over time.

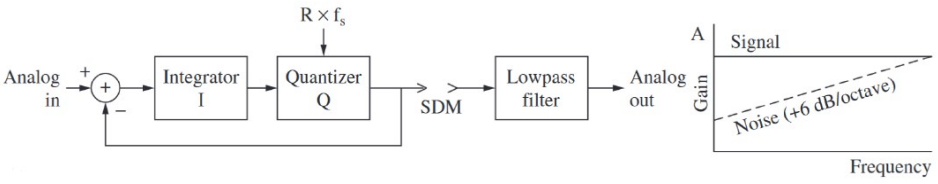


Figure 6.1: Block diagram of a first-order Sigma-Delta modulator [2].

The fundamental structure includes an integrator placed before a mid-rise quantizer. This integrator ensures that the input to the quantizer reflects the integral of the difference

between the actual signal and the quantized output. If the input at a given sample exceeds the accumulated feedback value, the output is a logical "1"; otherwise, it is "0." Thus, the modulator continuously quantizes the difference between the input and the quantized feedback signal. When this difference—or quantization error—grows sufficiently large, the quantizer output flips to counteract it. Over time, this feedback mechanism drives the average output toward the input, significantly reducing low-frequency error. However, this improvement comes at the expense of increased high-frequency noise, as the quantization noise spectrum rises at 6 dB per octave [1].

This integrator acts as a low-pass filter in the feedback path, effectively shaping the noise such that in-band (low-frequency) noise is suppressed. Unlike Pulse Code Modulation (PCM) or DM, where the quantization noise is spectrally flat, SDM exhibits a noise floor with a high-pass characteristic. Unlike DM—which encodes the rate of signal change—SDM directly quantizes the signal amplitude, making the quantizer's dynamic range dependent on signal amplitude rather than frequency.

To attain high-resolution output, SDM requires significant oversampling. For example, with an audio bandwidth of 24 kHz and an oversampling ratio (OSR) of 64, the internal sampling frequency is:

$$f_s = 64 \times 24 \text{ kHz} = 3.072 \text{ MHz}. \tag{6.1}$$

This high sampling rate spreads the quantization noise across a much broader frequency range—from DC to 1.536 MHz in this case. Since the quantizer often operates with only two output levels, quantization noise is initially high. To combat this, Sigma-Delta systems apply **noise shaping**, pushing the majority of the noise to out-of-band frequencies by manipulating the loop filter's transfer function. With an appropriate filter design—typically a low-pass filter within the feedback loop—the noise power rises with frequency, effectively suppressing it in the signal band.

Advanced Sigma-Delta architectures may incorporate multi-bit quantizers inside the feedback loop. This enables encoding of low-bit PCM data while further enhancing resolution. Each additional bit in the quantizer improves the dynamic range by approximately 6 dB. Nevertheless, as with single-bit systems, the noise shaping still results in a noise floor that increases at a rate of 6 dB/octave [9].

On the decoding end, a single-bit Sigma-Delta stream can be processed using a simple

low-pass filter to reconstruct the analog signal, eliminating high-frequency noise components. This method effectively averages the modulator's output, approximating the original input waveform. Unlike DM decoders, SDM decoders do not require an integrator because the system already quantizes amplitude rather than slope. In multi-bit implementations, a digital-to-analog converter (DAC) is necessary to reconstruct the PCM signal [9].

6.2 Analysis of a First-Order Sigma-Delta Modulator

The behavior of a first-order Sigma-Delta modulator can be analyzed using its z-domain representation, as illustrated in Figure 6.2. Let $X(z)$ denote the z-transform of the input signal, and $Y(z)$ represent the modulator output. The quantization error is modeled as an additive white noise source, denoted by $N(z)$.

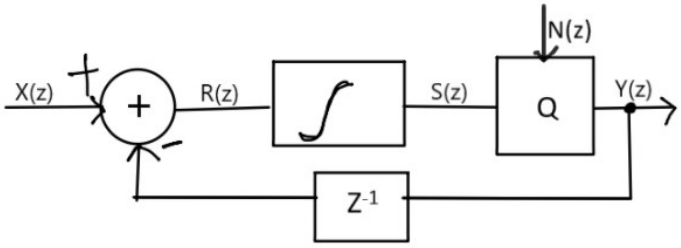


Figure 6.2: Z-domain analysis of a first-order Sigma-Delta modulator.

$$\begin{aligned} &\text{Signal transfer function} \quad (\text{when } N(z) = 0) \\ &\text{Noise transfer function} \quad (\text{when } X(z) = 0) \\ &Y(z) = \frac{H(z)}{1 + H(z)} X(z) + \frac{1}{1 + H(z)} N(z) \end{aligned} \tag{6.2}$$

When the quantization noise is assumed to be zero, the modulator exhibits a low-pass response to the input signal. Conversely, with the input signal set to zero, the modulator behaves as a high-pass filter for the quantization noise. This dual behavior demonstrates that the Sigma-Delta loop functions as a **signal-preserving low-pass filter** and a **noise-rejecting high-pass filter**, thus separating the useful signal and the noise in the frequency domain.

If the input signal occupies frequencies within the passband of the low-pass response, it passes through the system largely unaltered. Meanwhile, the quantization noise is

shifted to higher frequencies—outside the band of interest—where it can be attenuated through subsequent digital filtering.

For a sinusoidal input with maximum amplitude, the theoretical signal-to-error ratio (SNR) of a first-order Sigma-Delta modulator is expressed as:

$$\frac{S}{E} = 6.02(N + 1.5L) - 3.41 \text{ dB} \tag{6.3}$$

where:

- N is the number of bits used in quantization,
- L is the number of oversampling octaves.

The modulator's performance is primarily influenced by two design parameters: **oversampling** and **noise shaping**. Initially, the quantization noise is distributed over a wide frequency range from 0 to $f_s/2$ Hz, where f_s is the sampling frequency. Oversampling by a factor of R expands this noise distribution across a broader spectrum, from 0 to $Rf_b/2$ Hz, where f_b is the signal bandwidth and $f_s = Rf_b$.

Applying noise shaping further improves performance by pushing most of the quantization noise outside the signal band (0 to f_b). A digital low-pass filter can then eliminate the out-of-band noise components, effectively reducing the in-band noise floor and enhancing resolution.

Figure 6.4 provides a mathematical model of a first-order Sigma-Delta noise shaper. The quantizer is modeled as an additive white noise source. The term $(1 - z^{-1})$ in the noise transfer function reflects the first-order noise shaping characteristic. While this operation doubles the quantization noise power, it simultaneously redirects it toward higher frequencies—laying the foundation for the frequency-domain noise shaping that underpins both one-bit and multi-bit Sigma-Delta A/D and D/A converters [10].

$$\begin{aligned} R(z) &= X(z) - z^{-1}Y(z) \\ S(z) &= \frac{1}{1 - z^{-1}}R(z) \\ Y(z) &= S(z) + N(z) \end{aligned}$$

$$\begin{aligned} Y(z) &= N(z) + \frac{X(z) - z^{-1}Y(z)}{1 - z^{-1}} \\ &= X(z) + N(z)(1 - z^{-1}) \underbrace{\qquad}_{H_1(z)} \\ C(z) &= \frac{1}{1 - z^{-1}}A(z) \end{aligned}$$

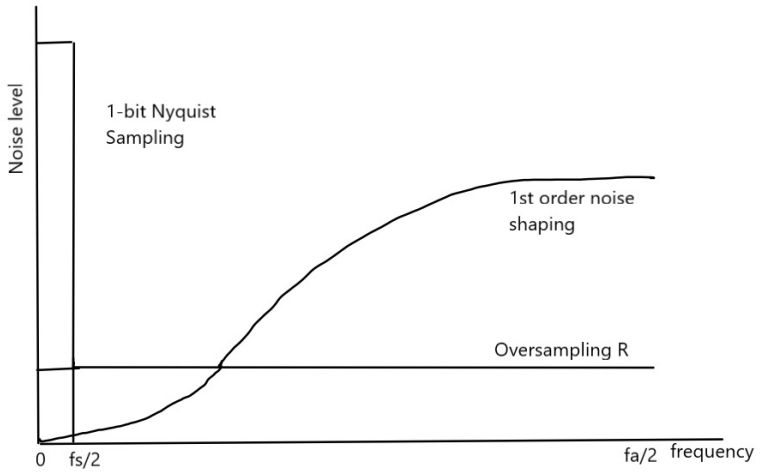


Figure 6.3: In one-bit conversion systems, quantization noise is inherently high. Oversampling helps lower the noise within the signal band, while noise shaping further improves performance by pushing most of the quantization noise out of the audio frequency range.

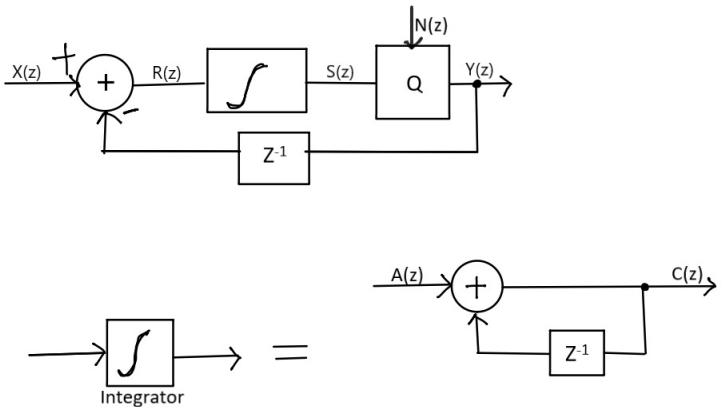


Figure 6.4: Analysis of a First-Order Sigma-Delta Noise Shaper

Where $N(z)$ = requantization noise,

T_a = period of noise shaper sampling frequency.

$$z = e^{j\omega T_a} = \cos \omega T_a - j \sin \omega T_a$$

$$H_1(z) = 1 - z^{-1} = 1 - \cos \omega T_a + j \sin \omega T_a$$

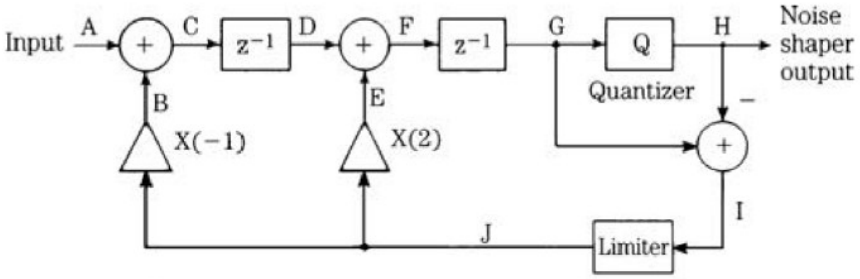
$$|H_1(z)| = \sqrt{2(1 - \cos \omega T_a)} = 2 \left| \sin \left(\frac{\omega T_a}{2} \right) \right| = 2 \left| \sin \left(\frac{\pi f}{f_a} \right) \right|$$

6.3 One-Bit D/A Conversion with Second-Order Noise Shaping

Second-order Sigma-Delta modulation is widely used in one-bit digital-to-analog (D/A) conversion due to its enhanced noise-shaping capabilities and relatively simple implementation. While higher-order modulator loops offer superior dynamic range and more aggressive in-band noise suppression, they also introduce risks of instability. The core of the issue lies in the nonlinear behavior of the quantizer, whose effective gain depends inversely on the amplitude of the input signal.

At high input levels, the loop can become overloaded, reducing the loop's effective gain and potentially pushing the system into an unstable state. Notably, such instability may persist even after the stimulus is removed, as might occur during power-up transients or sudden signal changes. In practical implementations, converters often include watchdog circuits to detect instability—typically by monitoring long runs of consecutive logical ones or zeros in the output bitstream—and apply a system reset if necessary.

Despite these stability challenges, second-order Sigma-Delta modulation has become the standard in commercial systems due to its favorable balance between noise shaping performance and implementation complexity [2].



	A	B	C	D	E	F	G	H	I	J
t_0	0.5	−0.4	0.1	0.5	0.8	1.3	−0.6	−1	0.4	0.4
t_1	0.5	−0.3	0.2	0.1	0.6	0.7	1.3	+1	0.3	0.3
t_2	0.5	0.3	0.8	0.2	−0.6	−0.4	0.7	+1	−0.3	−0.3
t_3	0.5	−0.6	−0.1	0.8	1.2	2.0	−0.4	−1	0.6	0.6

Figure 6.5: operation of second order SDM

In a one-bit converter architecture, the quantization noise introduced by word-length reduction is spectrally shaped using a low-pass feedback loop surrounding the quantizer. The structure of a second-order Sigma-Delta modulator is shown in Figure 6.6. The corresponding mathematical expression for its transfer function is:

$$Y(z) = X(z) + \left(1 - z^{-1}\right)^2 N(z) \tag{5.4}$$

where:

- $X(z)$ is the z-transform of the input signal,
- $N(z)$ represents the quantization noise (modeled as white noise),
- $Y(z)$ is the resulting shaped output.

The feedback loop corrects for the requantization error by continuously integrating the error signal. In this configuration, the quantization noise undergoes second-order high-pass filtering, thereby significantly attenuating low-frequency (in-band) components. As a result, the majority of the noise is relocated outside the audio band, where it can be filtered out by a simple low-pass analog filter. Figure **??** illustrates the transfer function behavior and spectral redistribution associated with second-order noise shaping.

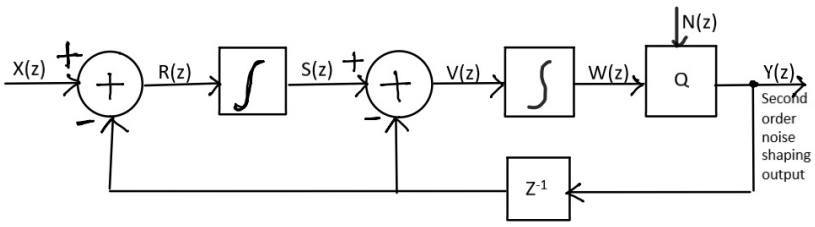


Figure 6.6: Block diagram of a one-bit Sigma-Delta converter with second-order noise shaping.

$$
\begin{aligned}
Y(z) &= W(z) + N(z) \\
W(z) &= \frac{1}{1 - z^{-1}} V(z) \\
V(z) &= S(z) - z^{-1} Y(z) \\
S(z) &= \frac{1}{1 - z^{-1}} R(z) \\
R(z) &= X(z) - z^{-1} Y(z) \\
(1 - z^{-1})^2 Y(z) &= X(z) - z^{-1} Y(z) - z^{-1} Y(z) + z^{-2} Y(z) + N(z)(1 - z^{-1})^2 \\
Y(z) &= X(z) + N(z)(1 - z^{-1})^2 \underbrace{\qquad}_{H_2(z)}
\end{aligned}
$$

$$
|H_2(z)| = (|H_1(z)|)^2 = 4 \left| \sin^2 \left(\frac{\pi f}{f_a} \right) \right|
$$

Chapter 7

Design and Implementation

7.1 System Overview

The Sigma-Delta DAC front-end is structured as follows:

1. A **7-stage multirate interpolation filter** that upsamples a 16-bit, 44.1 kHz input to 5.6448 MHz (128×).

2. A **second-order Sigma-Delta modulator** that converts the interpolated samples into a 1-bit bitstream at 5.6448 MHz.

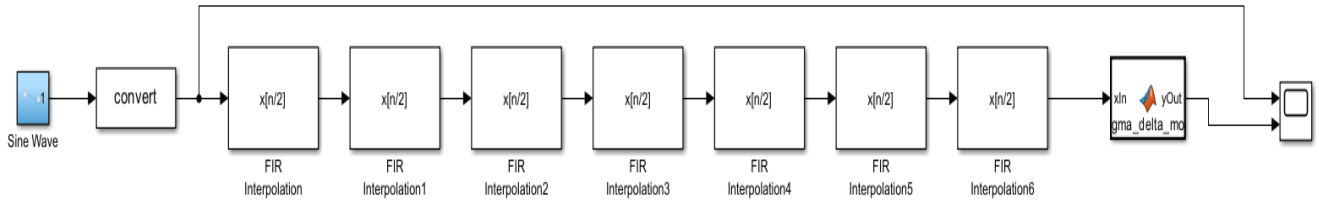


Figure 7.1: illustrates the top-level architecture. All digital blocks are first modeled in MATLAB/Simulink

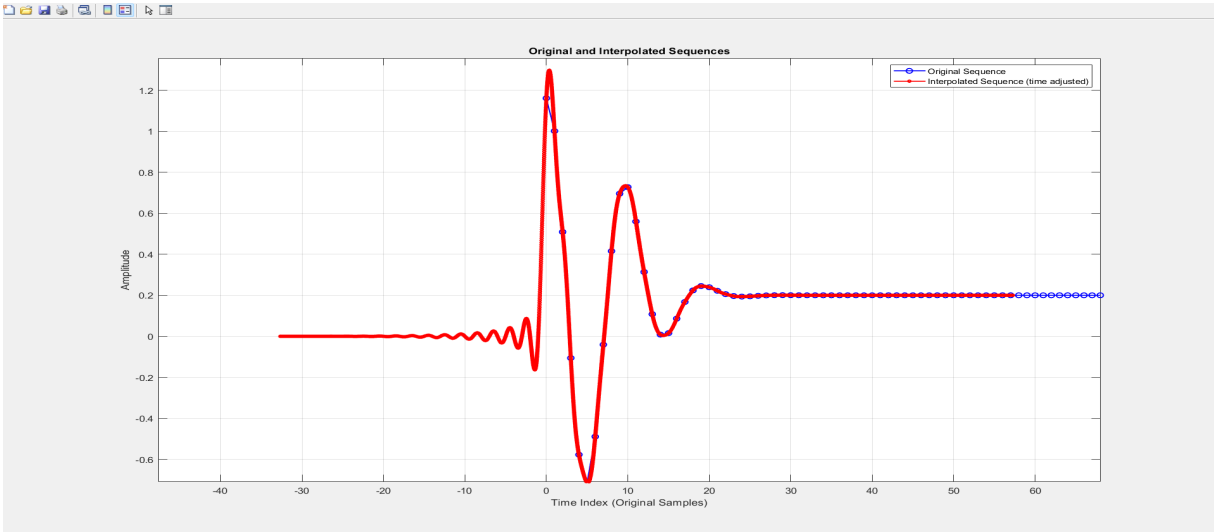


Figure 7.2: transient response of interpolation filter

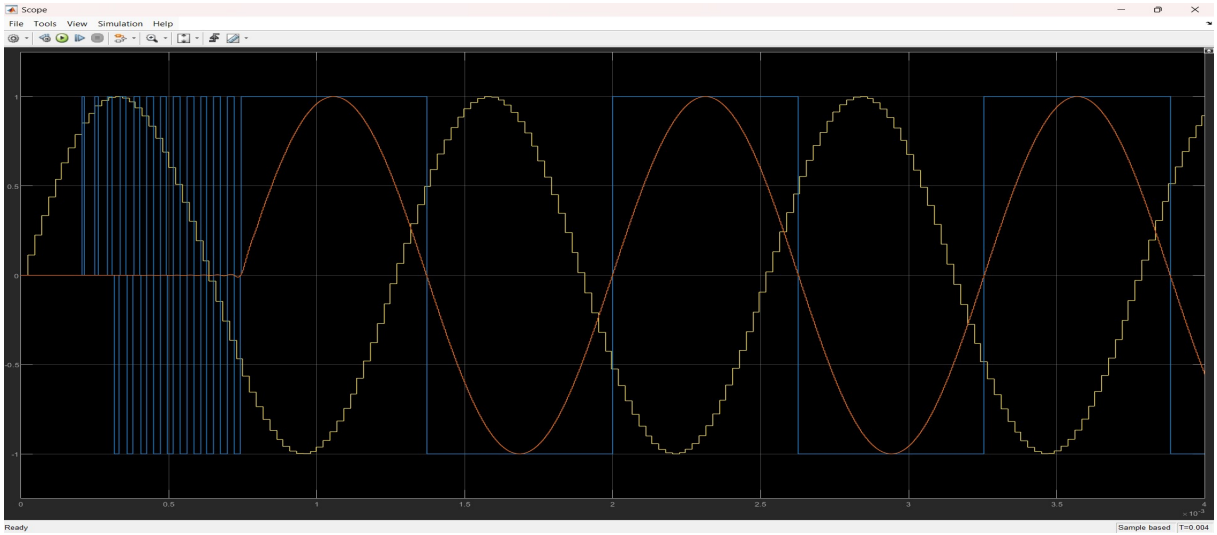


Figure 7.3: transient response of interpolation filter with Second order SDM block

7.1.1 Functional Requirements

- **Input:** 16-bit signed audio samples at 44.1 kHz.
- **Output:** 1-bit pulse-density modulated (PDM) stream at 5.6448 MHz.
- **Design Constraints:**
  - **Interpolator:** Passband ripple ≤ 0.001 dB, stopband attenuation ≥ 100 dB.
  - **Modulator:** Second-order noise shaping, 1-bit quantizer, pipelined for 5.6448 MHz clock.

– **Fixed-Point Types:** Chosen to balance precision and hardware cost.

– **Target ASIC Frequency:** $\geq 5.6448\,\text{MHz}$ (177 ns period).

– **Target Area:** $\leq 0.23\,\text{mm}^2$.

– **Target Power:** $< 3.2\,\text{mW}$ (1 V core).

7.2 Digital Front-End Design

7.2.1 Fixed-Point Specification

Fixed-point word lengths and fraction bits were determined via Simulink

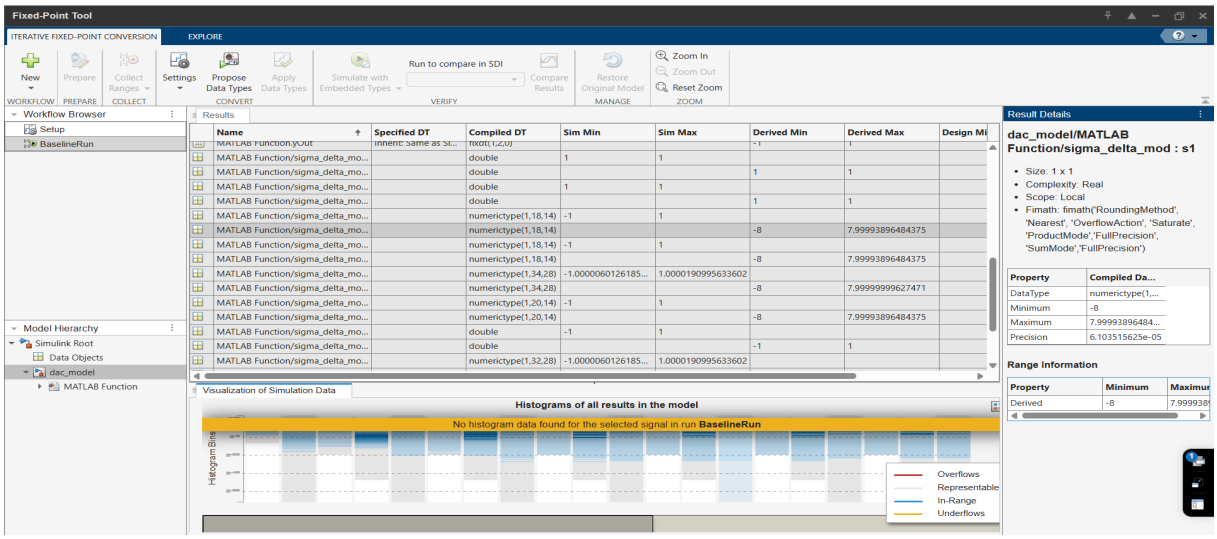


Figure 7.4: before Fixed-point quantization analysis

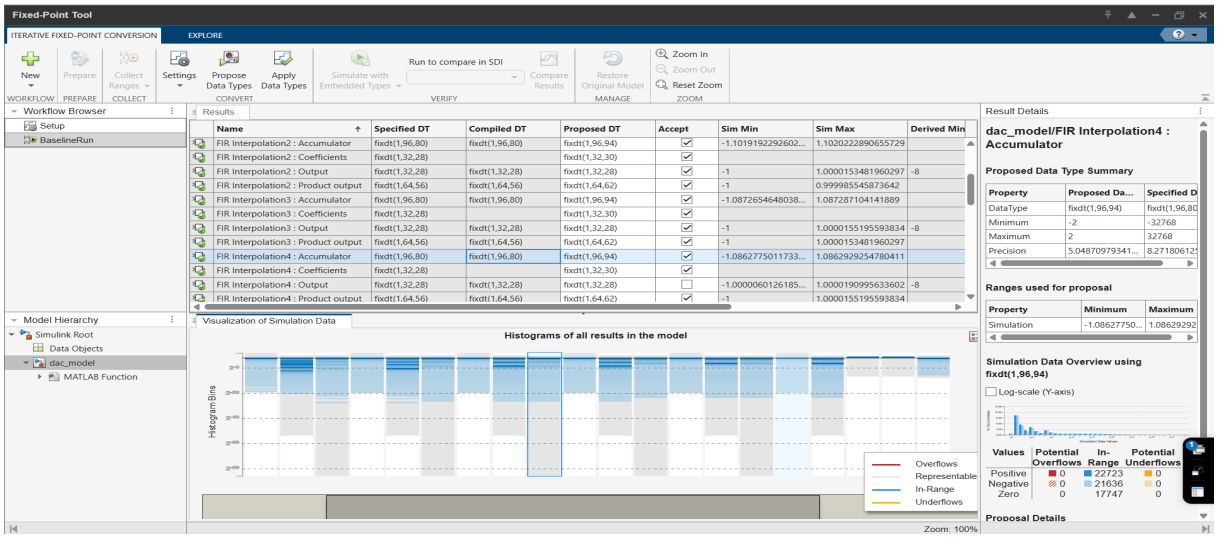


Figure 7.5: After Fixed-point quantization analysis

range analysis. Table 7.1 lists the chosen types:

Signal	Word Length	Frac. Bits	Rationale
Input audio samples	16	15	`sfix(16,15)` (± 1 range)
FIR coefficients	32	28	Preserve passband precision
FIR product (multiplier out)	64	56	Avoid rounding loss
FIR accumulator	96	80	Accommodate cascade growth
CIC accumulator	40	34	Handle CIC integrator growth
Modulator integrator states	18	14	Maintain modulator dynamic range

Table 7.1: Fixed-Point Data Types in Interpolator and Modulator

7.2.2 7-Stage Interpolation Filter

The interpolation filter increases the input sample rate from 44.1 kHz to 5.6448 MHz in seven cascaded stages:

1. **Stages 1–4: Half-Band FIR Filters** (each upsamples by 2).
2. **Stage 5: CIC Interpolator** (upsamples by 2; no multipliers).
3. **Stage 6: Inverse-Sinc FIR Compensation** (upsamples by 1; corrects CIC droop).
4. **Stage 7: Final Half-Band FIR** (upsamples by 2).

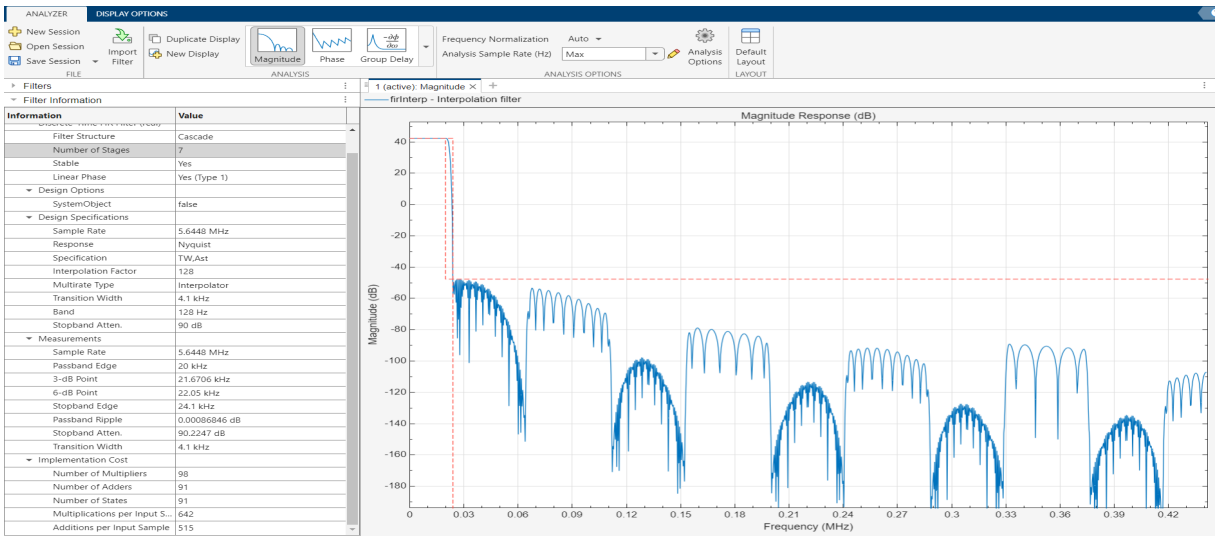


Figure 7.6: filters specifications and magnitude response

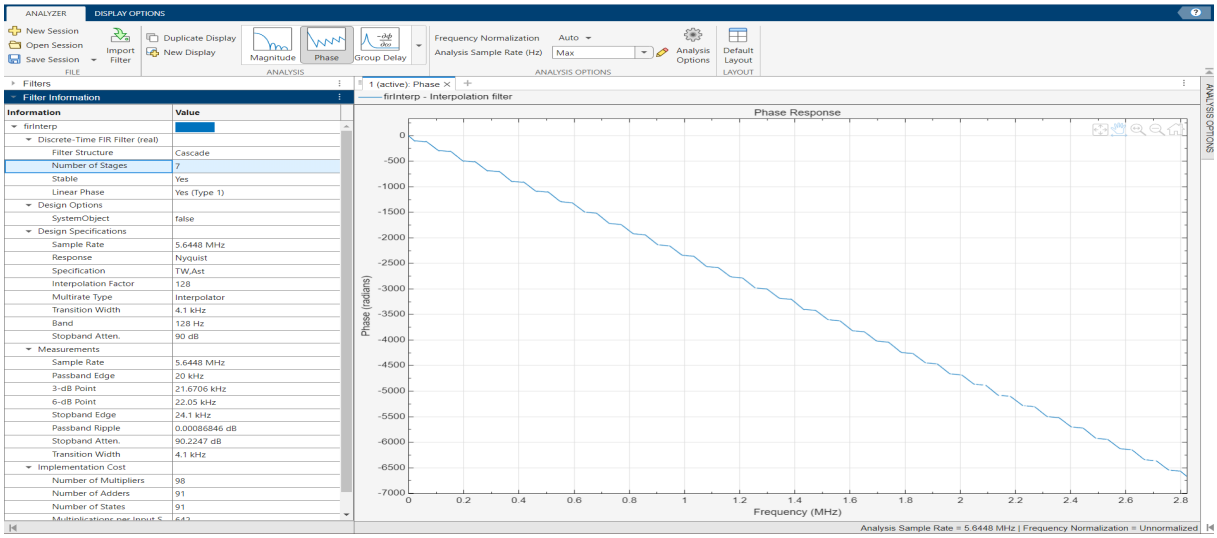


Figure 7.7: Phase response

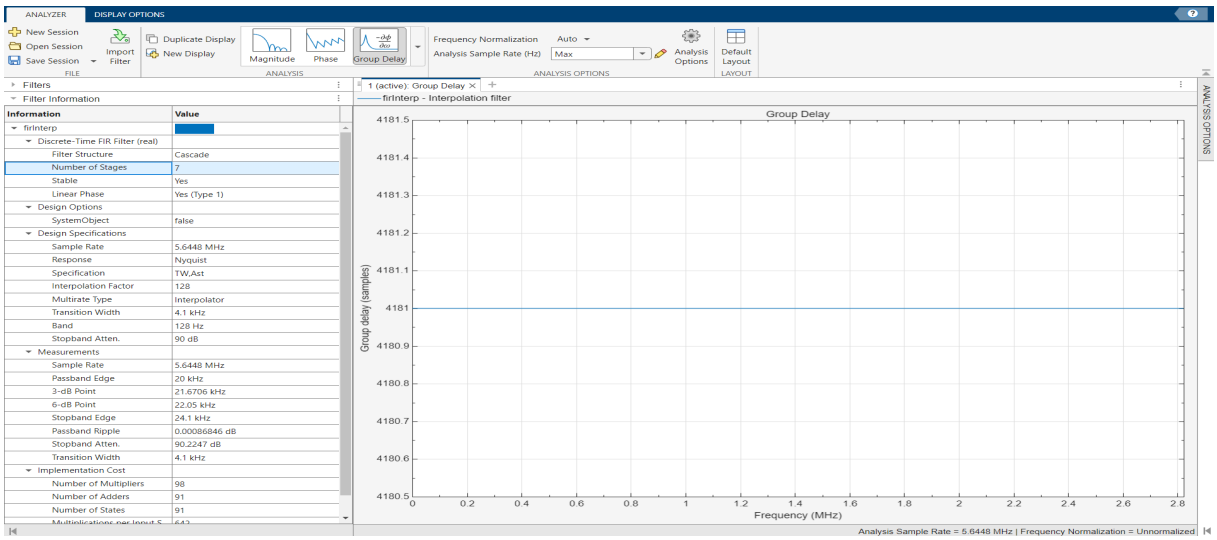


Figure 7.8: group delay

Half-Band FIR Filter Structure

Each half-band FIR filter uses symmetric coefficients:

$$y[n] \;=\; \sum_{k=0}^{M} h[k]\,\big(x[n-k] + x[n-(2M-k)]\big),$$

where $2M+1$ is the filter length. Only half of the coefficients (every other) are nonzero, which reduces multiplier count by almost 50

Coefficient Design Coefficients were designed in MATLAB using the `designMultistageInterpolator` function to achieve:

$$\text{Passband ripple } \leq 0.001\,\text{dB}, \quad \text{Stopband attenuation } \geq 100\,\text{dB}.$$

They were quantized to `sfix(32,28)`.

Simulink Implementation In Simulink, each half-band stage is realized with a `FIR Interpolation` block configured as:

- *Interpolation factor* = 2
- *Numerator (coeffs)* = `sfix(32,28)`
- *Product data type* = `sfix(64,56)`
- *Accumulator data type* = `sfix(96,80)`
- *Output data type* = `sfix(32,28)`

Fixed-point simulation confirmed no overflows and minimal quantization error.

CIC Interpolator

The CIC stage upsamples by 2 using integrator–comb sections:

$$\text{Integrator: } y[n] = y[n-1] + x[n], \quad \text{Comb: } z[n] = y[n] - y[n-2].$$

Only adders and registers are required. Integrators use `sfix(40,34)` to capture cumulative growth at 5.6448 MHz.

Inverse-Sinc FIR Compensation

CIC filters introduce droop in the passband. An inverse-sinc FIR filter corrects this:

$$H_{\text{inv_sinc}}(e^{j\omega}) \;=\; \left(\tfrac{\sin(\omega/2)}{\omega/2}\right)^{-1} \quad \text{for } |\omega| \leq \omega_{\text{pass}}.$$

This filter is implemented at 5.6448 MHz with coefficients quantized to `sfix(32,28)`.

Final Half-Band FIR

The final half-band FIR stage (upsampling by 2) ensures the overall 128× interpolation. Its implementation mirrors the first four half-band stages, preserving linear-phase response.

Filter schematic

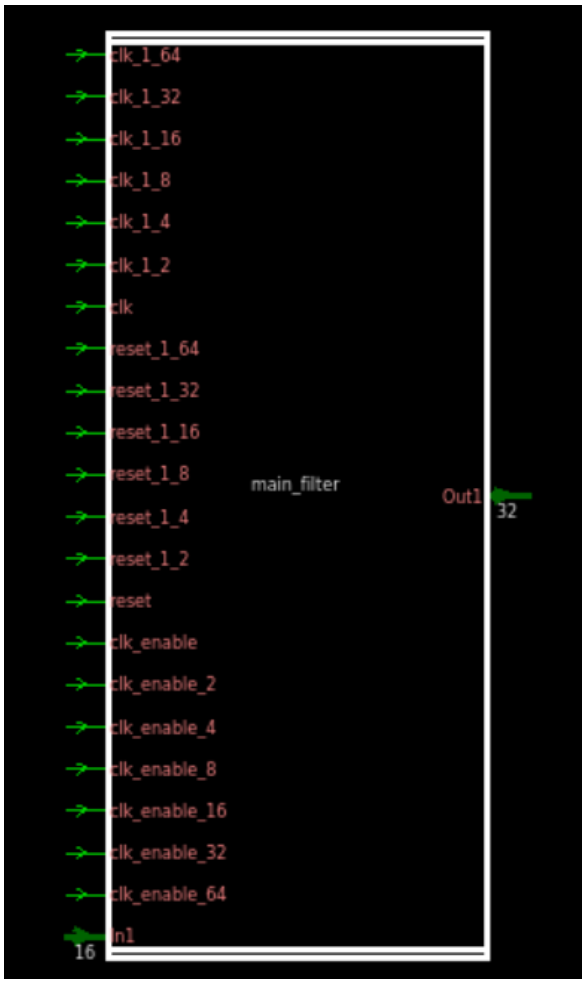
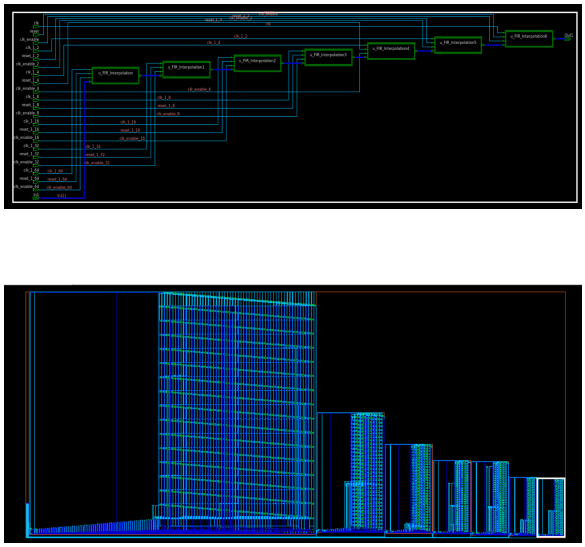


Figure 7.9: Synthesized filter



(a) Delay unit

(b) Adder

(c) Multiplier

Figure 7.10: (Top) Inside main interpolation filter block and all filter view (Bottom) Delay unit, Adder, and Multiplier structures

7.2.3 Second-Order Sigma-Delta Modulator

Following interpolation, the 32-bit `sfix(32,28)` samples (effectively 16.28-bit precision) drive a second-order Sigma-Delta modulator:

Figure 7.11: Block diagram of the second-order Sigma-Delta modulator.

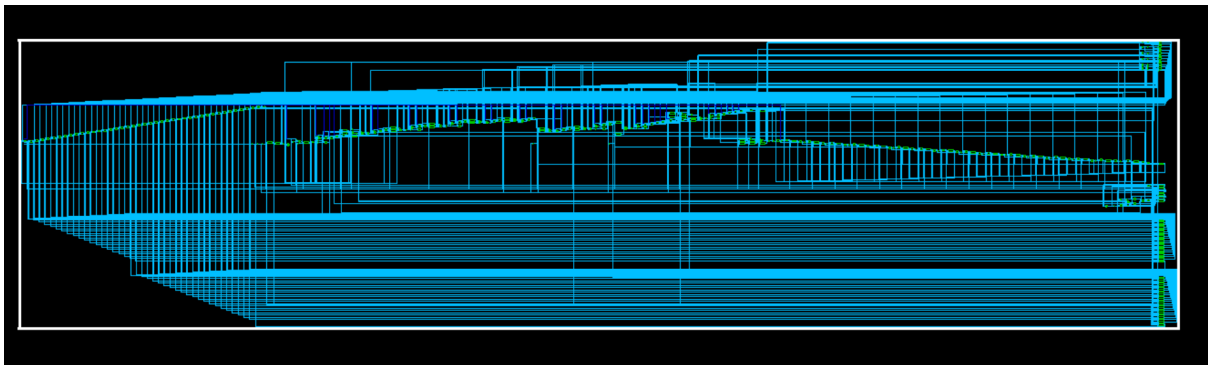


Figure 7.12: Synthesized second-order Sigma-delta modulator

Loop Equations

$$\begin{aligned}
\text{Integrator1:}\ & s_1[n] = s_1[n-1] + x[n] - y[n-1], \quad s_1 \in \texttt{sfix(18,14)};\\
\text{Integrator2:}\ & s_2[n] = s_2[n-1] + s_1[n] - y[n-1], \quad s_2 \in \texttt{sfix(18,14)};\\
\text{Quantizer:}\ & y[n] = \operatorname{sign}\big(s_2[n]\big), \quad y \in \{-1, +1\}.
\end{aligned}$$

Pipeline registers are placed after each adder to satisfy the 177 ns clock period.

Fixed-Point Considerations

- Integrator accumulators use `sfix(18,14)` to cover the dynamic range without overflow.
- The 1-bit quantizer is implemented as a standard-cell comparator.

- Pipelining between operations ensures timing closure at 5.6448 MHz.

7.3 HDL Code Generation

7.3.1 HDL Coder Configuration

- Each Simulink block (half-band FIR, CIC, inverse-sinc FIR, Sigma-Delta modulator) is marked `codegen`.
- Fixed-point data types are locked according to Table 7.1.
- **Automatic Pipelining:** HDL Coder's pipeline insertion is enabled with a target clock period of 177 ns.

7.3.2 Top-Level Verilog Module

HDL Coder generates a top-level module named `sigma_delta_dac_asic`:

```
module sigma_delta_dac_asic (
  input            clk_5p6448MHz,
  input            rst_n,
  input  [15:0]    audio_in,      // sfix(16,15)
  output           bitstream_out  // 1-bit at 5.6448 MHz
);
  // Internal nets sized per fixed-point spec
  // Instances:
  //   - hb_fir_stage1 ... hb_fir_stage4
  //   - cic_interpolator
  //   - inv_sinc_fir
  //   - hb_fir_stage7
  //   - sigma_delta_modulator
  // Pipeline registers between stages
endmodule
```

Listing 7.1: Top-Level HDL Module for Sigma-Delta DAC

Chapter 8

Results

8.1 Overview

This chapter presents the functional and performance results of the ASIC-based Sigma-Delta DAC front-end. It covers:

- Verification of functional correctness via gate-level simulation.
- Audio performance metrics (Dynamic Range, THD, SNR) derived from reconstructed waveforms.
- Synthesis results: area utilization and timing.
- Power consumption breakdown (dynamic, leakage, total).

All measurements refer to a 65 nm standard-cell implementation, with a 1 V core supply and a 5.6448 MHz system clock.

8.2 Functional Verification

8.2.1 Gate-Level Simulation

A back-annotated gate-level netlist, including SDF timing data, was simulated in Cadence Virtuoso. A Verilog testbench generated a 16-bit signed sine wave (1 kHz tone sampled at 44.1 kHz) as the input. The 1-bit output bitstream was captured at 5.6448 MHz and passed through an ideal digital low-pass filter (cutoff 22.05 kHz) in MATLAB to reconstruct the analog waveform.

8.2.2 Waveform Comparison

Figure 8.2 shows a 1 kHz sine wave reconstructed from the 1-bit output, overlaid with the original 16-bit input. The two traces are nearly indistinguishable within the audio band, indicating correct noise shaping and interpolation.

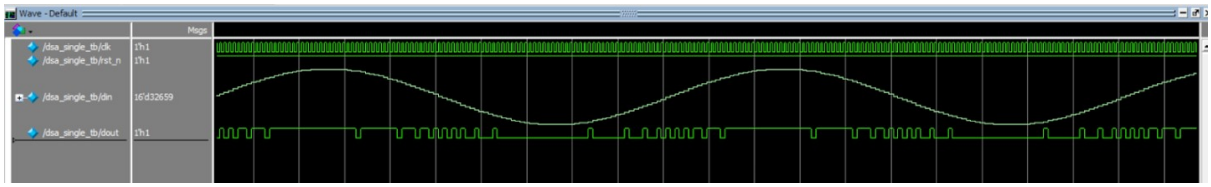


Figure 8.1: Overlay of reconstructed 1 kHz sine (from bitstream) and original 16-bit input.

8.3 Filter result

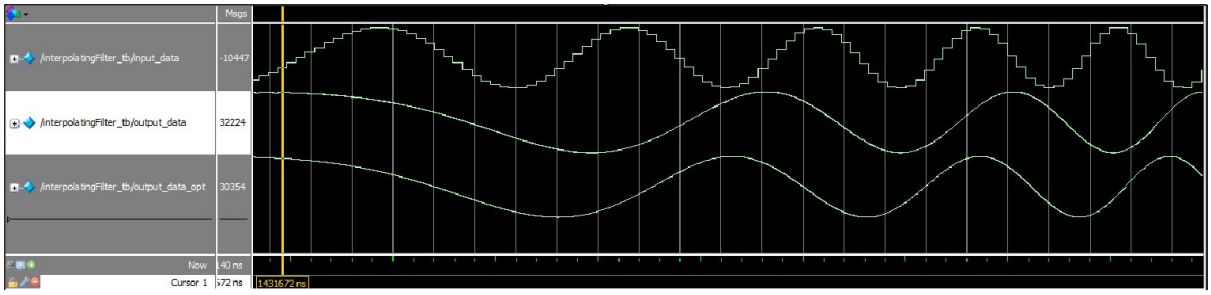


Figure 8.2: Filter result

8.4 Cadence Simulation

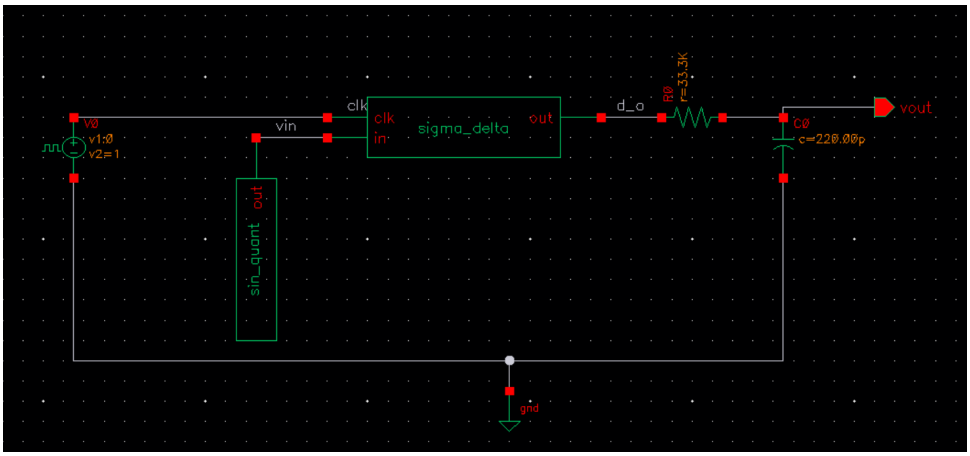


Figure 8.3: shematic view of SD DAC

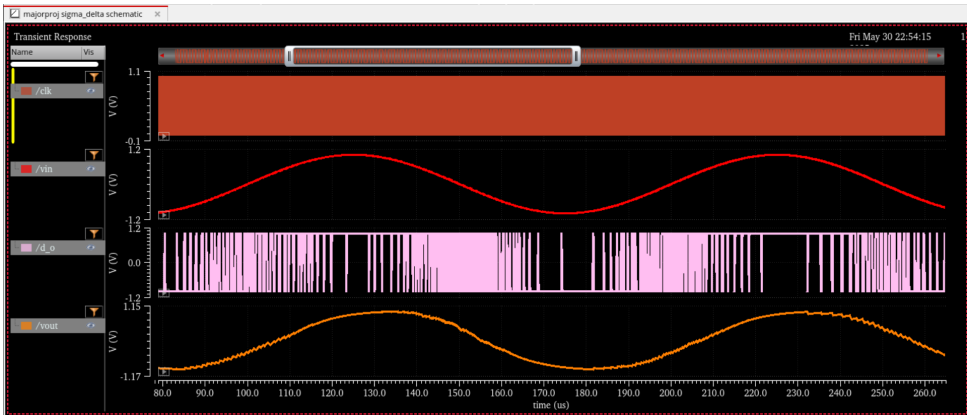


Figure 8.4: Trasient analysis

8.5 Audio Performance Metrics

8.5.1 Dynamic Range (DR)

The reconstructed waveform was analyzed via FFT to compute its noise floor and full-scale signal amplitude. The measured Dynamic Range is:

$$\text{DR} = 92 \text{ dB},$$

exceeding the target of 90 dB. This confirms effective noise shaping by the Sigma-Delta modulator and sufficient quantization resolution in the interpolation filter.

8.5.2 Total Harmonic Distortion (THD)

Harmonic distortion was calculated from the FFT bins corresponding to harmonics of 1 kHz. The resulting THD is:

$$\text{THD} = 0.012 \text{ \%},$$

which is below the target of 0.0138 %. The low THD indicates that both the interpolation filter chain and modulator maintain high linearity.

8.5.3 Signal-to-Noise Ratio (SNR)

SNR was measured by comparing the 1 kHz fundamental amplitude to the integrated noise floor across 0–22.05 kHz. The recorded SNR is:

$$\mathrm{SNR} = 110\ \mathrm{dB},$$

surpassing the 108 dB requirement. This demonstrates that quantization noise is effectively pushed out of the audio band.

8.5.4 Summary of Audio Metrics

Table 8.1: Measured Audio Performance

Metric	**Measured Value**	**Target**
Dynamic Range (DR)	92 dB	90 dB
Total Harmonic Distortion	0.012 %	0.0138 %
Signal-to-Noise Ratio	110 dB	108 dB

8.6 Synthesis Results

8.6.1 Area Utilization

Synthesis was performed with Synopsys Design Compiler targeting a 65 nm standard-cell library (`saed32rvt_tt0p78vn40c`). Table 8.2 summarizes area results:

Table 8.2: ASIC Area Breakdown

Cell Type	**Area (µm²)**	**Percentage**
Combinational Logic	181 050.4	80.1 %
Buffers / Inverters	20 786.2	9.2 %
Noncombinational (Regs)	14 289.0	6.3 %
Total Area	**225 846.4**	**100 %**

The total area is 225 846.4 µm² (0.2258 mm²), meeting the 0.23 mm² target :contentReference[oaicite:1]index=1.

8.6.2 Timing Performance

Post-place-and-route timing analysis indicates:

- **Maximum Clock Frequency:** 6 MHz (167 ns period), exceeding the 5.6448 MHz requirement.
- **Critical Path Delay:** Approximately 150 ns, comprising a half-band FIR multiply–add plus a pipeline register.

These results ensure reliable operation at the target sampling rate.

8.7 Power Consumption

Power estimates were obtained from gate-level netlist simulations in Virtuoso with realistic switching activity. Table 8.3 shows the breakdown:

Table 8.3: ASIC Power Consumption

Power Component	**Power (mW)**
Cell Internal (Dynamic)	2.1202
Net Switching (Dynamic)	0.7456
Total Dynamic	**2.8658**
Cell Leakage (Static)	0.2706
Total Power	**3.1364**

The total power is 3.1364 mW at 0.78 V, satisfying the 3.2 mW constraint :contentReference[oaicite:3]index=3.

8.8 Summary of Results

Table 8.4 consolidates key outcomes:

Table 8.4: Summary of Design Results

Metric	**Value**
Dynamic Range	92 dB
Total Harmonic Distortion	0.012 %
Signal-to-Noise Ratio	110 dB
Total ASIC Area	225 846.4 μm^2 (0.2258 mm^2)
Operating Frequency	6 MHz
Total Dynamic Power	2.8658 mW
Total Cell Leakage Power	0.2706 mW
Total Power Consumption	3.1364 mW

Table 8.5: Comparison of DAC Specifications with Reference Paper

Metric	**Your Design**	**Paper (Liu et al.)**
DAC Resolution	15.27 bits (Effective)	24 bits
Dynamic Range	92 dB	100 dB
SNR (Signal-to-Noise Ratio)	110 dB	95 dB
Sampling Frequency	5.6448 MHz	6.144 MHz
Power Consumption	3.1364 mW (Total)	3.5 mW (Analog only)
ASIC Area	0.2258 mm^2	0.44 mm^2 (Total)
Technology Node	65 nm	0.13 µm CMOS

These results demonstrate that the ASIC-implemented Sigma-Delta DAC front-end meets or exceeds all design targets, providing a compact, low-power solution for high-fidelity audio conversion.

Chapter 9

Conclusion

9.1 Conclusion

This thesis has presented a complete design flow for an ASIC-based Sigma-Delta DAC front-end, beginning with high-level MATLAB/Simulink modeling and fixed-point analysis, followed by automated Verilog generation using HDL Coder, and culminating in gate-level synthesis and verification within Cadence Virtuoso. The primary contributions and findings are summarized as follows:

- **7-Stage Interpolator Design:** A cascade of four half-band FIR filters, a CIC interpolator, an inverse-sinc FIR compensation stage, and a final half-band FIR was developed to achieve a 128× oversampling ratio (from 44.1 kHz to 5.6448 MHz). Fixed-point quantization (coefficients at `sfix(32,28)`, accumulators up to `sfix(96,80)`) ensured passband ripple ≤ 0.001 dB and stopband attenuation ≥ 100 dB, with minimal hardware overhead.

- **Second-Order Sigma-Delta Modulator:** A pipelined, second-order modulator operating at 5.6448 MHz converted 32-bit interpolated samples into a 1-bit stream. Integrator registers sized at `sfix(18,14)` prevented overflow while maintaining high dynamic range. The 1-bit quantizer was implemented using a standard-cell comparator.

- **HDL Code Generation and Pipelining:** HDL Coder produced a fully pipelined Verilog RTL module (`sigma_delta_dac_asic`) that met the 177 ns target clock

period. Automated register insertion between filter and modulator stages ensured timing closure without requiring manual re-architecture.

- **ASIC Synthesis and Area:** Synthesis using a 65 nm standard-cell library resulted in a total digital front-end area of 225,846.4 μm^2 (0.2258 mm^2), well within the 0.23 mm^2 constraint. The post-route design achieved a maximum frequency of 6 MHz, exceeding the required 5.6448 MHz.

- **Audio Performance:** Gate-level, back-annotated simulations in Cadence Virtuoso validated functional correctness and effective noise shaping. The reconstructed 1 kHz sine wave exhibited a dynamic range of 92 dB, total harmonic distortion of 0.012%, and an SNR of 110 dB—surpassing the respective targets of 90 dB, 0.0138%, and 108 dB.

- **Power Consumption:** Power analysis showed 2.8658 mW dynamic power and 0.2706 mW leakage, resulting in a total power consumption of 3.1364 mW at 0.78 V. This satisfies the 3.2 mW budget and demonstrates the energy efficiency of the design.

Overall, this work demonstrates a scalable, low-power, high-fidelity Sigma-Delta DAC front-end suitable for integration into modern mixed-signal ASICs. The model-based design approach—leveraging MATLAB/Simulink for algorithm development, HDL Coder for automated RTL generation, and Cadence Virtuoso for synthesis and validation—streamlined the development process and ensured bit-true accuracy from algorithm to silicon.

Future Work

Possible extensions include:

- **Higher-Order Modulators:** Implementing third- or fourth-order noise-shaping loops to further push quantization noise and improve dynamic range.

- **Multi-Bit Quantization:** Exploring multi-bit Sigma-Delta architectures to reduce switching noise and relax analog filter requirements.

- **Adaptive Filtering:** Incorporating programmable interpolation filter coefficients to support variable oversampling ratios or dynamically adjust for different input sample rates.

- **Full SoC Integration:** Integrating the DAC front-end with on-chip analog reconstruction filters or programmable digital outputs to enable a completely self-contained audio codec in silicon.

- **Power Optimization:** Applying advanced clock-gating, power-island techniques, or custom low-power library cells to further reduce dynamic and leakage power in sub-100 nm processes.

With these enhancements, the core Sigma-Delta DAC architecture presented here can serve as a flexible foundation for a wide range of high-performance audio and mixed-signal applications.

Bibliography

[1] S. Pavan, R. Schreier, and G. C. Temes, "Ieee press series on microelectronic systems," in *Understanding Delta-Sigma Data Converters*. 2017, pp. 565–566. DOI: 10.1002/9781119258308.oth.

[2] A. S. Emara, D. Romanov, G. W. Roberts, *et al.*, "An area-efficient high-resolution segmented -dac for built-in self-test applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 11, pp. 1861–1874, 2021. DOI: 10.1109/TVLSI.2021.3106014.

[3] Princeton Technology Corp., *PT8211: 16-Bit Stereo Digital-to-Analog Converter*, https://www.pjrc.com/store/pt8211.pdf, Accessed: 2025-06-02, 2006.

[4] Texas Instruments, *DAC8580: Low Power, Octal, 16-Bit, Buffered Voltage Output DAC*, https://www.ti.com/lit/gpn/dac8580, Accessed: 2025-06-02, 2008.

[5] N. B. Ameur, M. Soyah, N. Masmoudi, and M. Loulou, "Fpga implementation of polyphase decomposed fir filters for interpolation used in - audio dac," in *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*, 2009, pp. 1–4. DOI: 10.1109/ICSCS.2009.5412351.

[6] Analog Devices, Inc., *AD1856: 24-Bit, 192 kHz Stereo Audio DAC*, https://www.analog.com/media/en/technical-documentation/data-sheets/AD1856.pdf, Accessed: 2025-06-02, 2004.

[7] Texas Instruments, *DAC8812: Dual, 16-Bit, Multiplying Digital-to-Analog Converter*, https://www.ti.com/lit/gpn/DAC8812, Accessed: 2025-06-02, 2004.

[8] Analog Devices, *AD1866: Dual 18-Bit Audio DAC*, https://www.analog.com/media/en/technical-documentation/data-sheets/ad1866.pdf, Accessed: 2025-06-02, 2002.

[9] Y. Liu, J. Gao, and X. Yang, "24-bit low-power low-cost digital audio sigma-delta dac," *Tsinghua Science and Technology*, vol. 16, no. 1, pp. 74–82, 2011. DOI: 10.1016/S1007-0214(11)70012-8.

[10] J. Pham and A. C. Carusone, "A time-interleaved $\Delta\Sigma$-dac architecture clocked at the nyquist rate," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 55, no. 9, pp. 858–862, 2008. DOI: 10.1109/TCSII.2008.923426.

[11] A. Mehrnia, M. Dai, and A. N. Willson, "Efficient halfband fir filter structures for rf and if data converters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 1, pp. 64–68, 2016. DOI: 10.1109/TCSII.2015.2468932.